

Señores
Superintendencia de Industria y Comercio
División de Nuevas Creaciones
E. S. D.

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO



No. 03-046838- -00008-0000

Fecha: 2008-05-20 15:21:06 Dep. 2020 NUECREACION
Tra. 2 PATENTES Eve: 1 REGDEPOSITO
Act. 412 REPOSPRESRECU Folios: 474

Ref: Expediente No. 03-046.838

Solicitud de registro de patente de invención denominada "COMPUTACIÓN DISTRIBUIDA BASADA EN IDENTIDAD PARA RECURSOS DE ARTEFACTOS"

Solicitante: MICROSOFT CORPORATION.

Dario Cárdenas Navas, identificado como aparece al pie de mi firma y obrando en mi condición de Representante Legal en Bogotá para todos los asuntos relativos a la Propiedad Industrial de la sociedad **MICROSOFT CORPORATION**, organizada y existente de conformidad con las leyes de los Estados Unidos de América, y domiciliada en Microsoft Way, Redmond, Washington 98052, atentamente manifiesto a usted que interpongo recurso de reposición contra la Resolución No. 11341, dictada por el señor Superintendente de Industria y Comercio, mediante la cual se niega la patentabilidad de la solicitud Colombiana de la referencia.

Objeto del Recurso.

Solicito mediante el presente memorial, ordenar a quien corresponda:

1. **Revocar** la Resolución No. 11341 fechada 17 de Abril de 2008 y notificada el 13 de Mayo de 2008.
2. Que en su lugar se **conceda** el Registro a la solicitud de patente titulada **COMPUTACIÓN DISTRIBUIDA BASADA EN IDENTIDAD PARA RECURSOS DE AREFACTOS**.
3. Que de lo contrario **conceda un segundo examen de fondo** para la sustentación del invento en referencia.

Fundamentos de la Providencia recurrida

Señala la providencia recurrida que se niega el privilegio de patente de invención pues la misma no cumple con lo dispuesto en los Artículos 14, 15e), 21 y 30 de la Decisión 486 de la Comunidad Andina.

Fundamentos del Recurso.

1. Señala el Despacho que la solicitud de patente en referencia no es patentable puesto que no cumple con lo dispuesto en los Artículos 14, 15e), 21 y 30 de la Decisión 486 de la Comunidad Andina.

Artículo 14.- Los Países Miembros otorgarán patentes para las invenciones, sean de producto o de procedimiento, en todos los campos de la tecnología siempre que sean nuevas, tengan nivel inventivo y sean susceptibles de aplicación industrial.



Artículo 15e).- No se considerarán invenciones:

- e) los programas de ordenadores o el soporte lógico, como tales;

Artículo 21.- Los productos o procedimientos ya patentados, comprendidos en el estado de la técnica, de conformidad con el artículo 16 de la presente Decisión, no serán objeto de nueva patente, por el simple hecho de atribuirse un uso distinto al originalmente comprendido por la patente inicial.

Artículo 30.- Las reivindicaciones definirán la materia que se desea proteger mediante la patente. Deben ser claras y concisas y estar enteramente sustentadas por la descripción.

Las reivindicaciones podrán ser independientes o dependientes. Una reivindicación será independiente cuando defina la materia que se desea proteger sin referencia a otra reivindicación anterior. Una reivindicación será dependiente cuando defina la materia que se desea proteger refiriéndose a una reivindicación anterior. Una reivindicación que se refiera a dos o más reivindicaciones anteriores se considerará una reivindicación dependiente múltiple.

2. Señala el Despacho que la solicitud en referencia es un software (soporte lógico) como tal. Sin embargo respetuosamente manifiesto que no se encuentra sustentado ni motivado en ningún aparte de la Resolución ahora recurrida, por que se considera que el invento solicitado es un software (soporte lógico) como tal y lo que no se entiende como software (soporte lógico) como tal.

Entendemos que para el Despacho y el solicitante la concepción de software (soporte lógico) es diferente. Aclaramos que el solicitante y para las personas versadas en la materia consideran que el objeto aquí patentado es un invento que si bien tiene relación con elementos de software (soporte lógico) no es el software (soporte lógico) como tal.

3. Me permito señalar al Despacho que la norma Andina no prohíbe el registro de *software (soporte lógico)*, sino el registro del software (soporte lógico) como tal. Elemento que aún no ha sido definido por ese Despacho, ni por la doctrina.
4. Se anota que la norma andina no señala que el software no es patentable en su concepto restringido y limitado. Es decir, la prohibición del registro de software se refiere al **software como tal**.

Es decir, la norma claramente señala que la materia no patentable es el software como tal, lo que indica que el software podrá ser utilizado como una herramienta tecnológica para desarrollar, crear y obtener nuevos inventos. Existe, y ha sido uniformemente admitido por oficinas tales como la de los Estados Unidos, la Oficina de Patentes Europea, la oficina Japonesa y otras como la Oficina de marcas China, que existe una diferencia clara entre un software como tal (soporte lógico como tal), o un software *per se* (soporte lógico *per se*) e inventos implementados por computador.

Para tal efecto, respetuosamente me permito solicitar al Despacho que se sirva aclarar qué entiende la Superintendencia de Industria y Comercio colombiana como *software como tal* (soporte lógico como tal) y si existe

en su opinión alguna diferencia con los inventos implementados por computador.

Adjuntamos como prueba del desarrollo de la problemática de la patentabilidad del software el documento emitido por AIPPI bajo la Q132 reporte y memorando cuya autenticidad podrá confirmar el examinador en la página www.aippi.org capítulo Questions & Committee Q 132.

RESUMEN

Las normas actuales sobre invenciones implementadas por ordenador en los sistemas de patentes más importantes son las siguientes:

- Según la práctica norteamericana, cualquier invención útil hecha por el hombre es patentable, incluyendo cualquier programa de ordenador y método comercial nuevo y no evidente.
- En Japón, las invenciones de programas de ordenador y los métodos comerciales implementados por programas de ordenador son patentables, si se define en las reivindicaciones una interacción específica con un recurso de hardware.
- La Oficina Europea de Patentes concede protección por patentes a invenciones relacionadas con programas de ordenador, incluyendo programas de ordenador relacionados con métodos comerciales, siempre y cuando tengan carácter técnico y realicen una contribución técnica.

Las anteriores Resoluciones de AIPPI Q133 y especialmente Q155 requieren un contenido técnico como un presupuesto de patentabilidad. La AIPPI reconoce como suficiente, el contenido técnico de una invención implementada en un sistema técnico (que es siempre el caso de un programa de ordenador), aunque la contribución del inventor no tenga un carácter técnico. De acuerdo con la AIPPI, los métodos comerciales nuevos e inventivos y otras innovaciones no técnicas deben ser patentables si se implementan por un ordenador.

La Unión Europea (UE) intentó una armonización de las normas relativas a invenciones relacionadas con programas de ordenador a través de una directiva europea. La directiva europea fue rechazada en julio de 2003. Los puntos más importantes de discusión durante el proceso legislativo en la Unión Europea estuvieron relacionados con la existencia de beneficios económicos de la protección por patentes para las invenciones relacionadas con programas de ordenador y sobre la patentabilidad constituiría un impedimento a la creatividad y al progreso.

Dos complejos revisores recientes del sistema de patentes norteamericano han sido efectuados por la Comisión de Comercio Federal de los Estados Unidos (FTC), y por las Academias de Ciencias Nacionales (NAS). Ambas han concluido que aunque hay numerosas reformas prácticas necesarias para hacer que el sistema de patentes norteamericano funcione de forma más efectiva, en general las patentes tienen un papel importante que jugar en la economía moderna. El informe NAS señala explícitamente que «los altos índices continuados de innovación sugieren que el sistema de patentes está funcionando bien y no requiere cambios fundamentales». Asimismo, las extensiones previstas efectuadas por la Unión Europea en la preparación de la propuesta de la Directiva parecen ir en la misma dirección.

Sin embargo, coexiste con la industria de programas de ordenador comerciales que se aprovecha del sistema de patentes una gran y creciente comunidad de software de código abierto, que desarrolla programas de ordenador principalmente sin inversión comercial. Aquellos que trabajan en esta comunidad podrían tener poca o ninguna necesidad de patentes. Sin embargo, el mundo económico del software de código abierto está cambiando. Fabricantes de hardware tales como IBM han adoptado software de código abierto y facilitan versiones libres de Linux OS y otro software libre, que trabajan con su hardware con el fin de mantenerse competitivos en costes.

Ese Despacho debe notar que este documento refleja la situación de la patentabilidad del software al **2006**, sin embargo desde esa época han existido desarrollos jurisprudenciales tales, que resultan, en este caso en particular por ejemplo en la concesión del registro de la patente en referencia tanto en el territorio **Mexicano** como en la **Unión Europea**.

Solicito que en virtud del contenido de los Artículos 52 a 57 del C.C.A. se valoren como pruebas de la patentabilidad de la solicitud que se tramita bajo el Expediente en referencia el documento anexo "Protección por Patentes de Invenciones relacionadas con Programas de Ordenador" emitido por AIPPI el 2 de Enero de 2006.

6. Igualmente me permito informar al Despacho que tanto el suscrito como la sociedad solicitante actúan en este procedimiento en buena fe y que no es intención de ninguno de los dos actores actuar de una manera diferente, la única intención es tener una discusión sobre la patentabilidad

del software reflejando las tendencias mundiales del tema en países cuya legislación es similar a la nuestra incluyendo las restricciones de patentabilidad similares a las incluidas en nuestra Decisión 486.

7. Vemos, que en este caso existe una interpretación de parte del Despacho, al clasificar la solicitud en referencia como *software (soporte lógico) como tal ó per se* y que no se ha abierto el foro de discusión respecto de la patentabilidad de invenciones implementados por computador.

Es importante anotar que esta patente ya fue **concedida a registro como invención bajo las jurisdicciones de Mexico y de la Unión Europea según documentos que ya reposan en el expediente, los cuales fueron radicados el pasado día 18 de Abril de 2008.**

Solicito que en virtud del contenido de los Artículos 52 a 57 del C.C.A. se valoren como pruebas de la patentabilidad de la solicitud que se tramita bajo el Expediente en referencia las decisiones concedidas bajo las legislaciones Mexicana y Europea.

8. Tal como se señala en el documento AIPPI, se ha diferenciado entre *software per se* e invenciones relacionadas con programas de ordenador así:

"La materia patentable en tales innovaciones son los nuevos aparatos con los programas y/o el método integrado, es decir, el nuevo mecanismo para alcanzar un fin específico, práctico, concreto y útil y no la implementación de un programa de ordenador específico. El código fuente del programa de ordenador se protege por propiedad intelectual (derecho de autor ó copyright) y no por patentes. La expresión específica del programa (por ejemplo, el listado ó el código específico) contra las copias mientras que la protección por patente protege los elementos de un nuevo método (por ejemplo, la funcionalidad) independientemente de la implementación del código específico de un programa. La protección que otorga la ley de propiedad intelectual (derecho de autor/copyright) otorga una protección fuerte contra la simple multiplicación (por ejemplo, la copia directa ó el uso no autorizado) de un producto de software. La patente protege contra la copia del aparato y los métodos que métodos que puedan utilizar un programa de ordenador. Además, la propiedad intelectual (derecho de autor/copyright) y los derechos de patente son de una naturaleza diferente. En la práctica, la expresión de una idea es protegible por la propiedad intelectual (derecho de autor/copyright) por el mero hecho de que se trata de una creación, por ejemplo la expresión de cualquier programa de ordenador original es protegible por la propiedad intelectual (derecho de autor/copyright). Por el contrario para que sea patentable, una invención debe ser nueva y no evidente (no obvia) y esta condición no es fácilmente cumplimentada por un producto de software. Por ejemplo, un programa de ordenador que meramente implementa una determinada especificación no es patentable si solamente se han utilizado para la implementación de técnicas de programación normales. Solamente pueden ser inventivas y definidas en una reivindicación de patente la materia nueva y no evidente de la especificación y/o implementaciones funcionales de la especificación de nuevas y no evidentes."

9. Como se profundizará en documentos que apoyan este recurso que el software (soporte lógico) *per se* es considerado como las líneas en un lenguaje de soporte lógico que incorporan la implementación de una invención.

Mientras que el método ó pasos de la presente solicitud, si bien pueden ser implementados por computador, pueden también ser implementados mediante un circuito electrónico por ejemplo.

10. De hecho, y en concordancia con lo anterior, el Artículo 27 del ADPIC señala lo siguiente:

Art. 27.- Materia patentable:

1. Sin perjuicio de lo dispuesto en los párrafos segundo y tercero, las patentes podrán obtenerse por todas las invenciones, sean de productos ó de procedimientos, en todos los campos de la tecnología, siempre que sean nuevas, entrañen una actividad inventiva y sean susceptibles de aplicación industrial. ... los derechos de patente se podrán gozar sin discriminación por el lugar de la invención, el campo de la tecnología ó el hecho de que los productos sean importados ó producidos en el país.
2. ...
3. Los Miembros podrán excluir asimismo de la patentabilidad:
 - a) Los métodos de diagnóstico, terapéuticos y quirúrgicos para el tratamiento de personas ó animales.
 - b) Las plantas y los animales excepto los microorganismos, y los procedimientos esencialmente biológicos para la producción de plantas ó animales, que no sean procedimientos no biológico ó microbiológicos. Sin embargo, los Miembros otorgarán protección a todas las obtenciones vegetales mediante patentes, mediante un sistema eficaz sui generis o mediante una combinación de aquellas y este. Las disposiciones de del apartado serán objeto de examen cuatro años después de la entrada en vigor del Acuerdo sobre OMC.

Esta reglamentación es un indicativo del consenso que al que se está llegando uniformemente a nivel internacional en relación con la patentabilidad de la materia tal como la definida en referencia. De hecho el Artículo 27 3) no hace una excepción en relación con aquellas invenciones que no son software *per se* y que son consideradas como invenciones a nivel mundial.

11. En relación con el caso en particular, a continuación me permito fundamentar el recurso en los siguientes hechos:

- (1) Mediante Resolución **No. 11341**, se niega la patentabilidad de la presente invención basándose nuevamente en la supuesta falta de claridad de las reivindicaciones, además de que no se considera como una invención de acuerdo con el artículo 15 e) de la Decisión 486, y de que se considera en forma equivocada que se pretende reivindicar un segundo uso de elementos ya conocidos (según el artículo 21 de la Decisión 486), tal y como se había manifestado en el Concepto Técnico **No. 586 (Oficio No. 4276)**. En vista de lo que establece la Resolución Impugnada, se puede apreciar que **a pesar de las modificaciones y de los argumentos presentados oportunamente como respuesta a dicho concepto técnico, no se le ha dado la suficiente importancia a los argumentos técnicos presentados por el solicitante** que demuestran en forma adecuada que las nuevas reivindicaciones 1 a 29 de la presente solicitud cumplen cabalmente con las exigencias del citado artículo 30 de la Decisión 486, y que también demuestran que la materia objeto de

dichas reivindicaciones 1 a 29, no solo resulta clara y precisa, sino que además, resulta patentable, toda vez que no se trata de materia que se pueda considerar como incluida en las excepciones de patentabilidad, o como materia no patentable, en vista de que para una persona medianamente versada en la materia técnica, la invención denegada resulta ser un avance en la técnica que representa una solución a problemas técnicos aún existentes.

En vista de lo anterior, mediante el presente escrito se pretende llamar la atención de ese Despacho a algunos hechos que resultan de particular importancia para demostrar que la materia objeto denegada, si cumple con las exigencias técnicas y legales para gozar del privilegio de patente solicitado. Por lo tanto, a continuación me permito sustentar desde el punto de vista técnico y legal los aportes realizados por la presente invención que la hacen novedosa, inventiva y con aplicación industrial.

- (2) En efecto, de acuerdo con la Resolución Impugnada, ese Despacho objetó nuevamente la claridad de la solicitud argumentando que *"Las reivindicaciones 1, 6, 7, 17 y sus dependientes son indefinidas porque no permiten establecer cual es la contribución técnica al arte. Las reivindicaciones 1, 6, 7, 17 son muy generales, porque se relacionan con un método, un medio legible por computador y un sistema cualquiera; además, junto con sus dependientes presentan como pasos de un método, como medios de un sistema una secuencia de procesamiento de datos ejecutada por computador. Dado lo anterior, se advierte que las mencionadas reivindicaciones no cumplen con el requisito de claridad, no obstante fueron evaluadas y calificadas."*

Sobre este particular, atentamente me permito manifestar a ese Despacho que con base en las modificaciones realizadas como respuesta al anterior Oficio emitido por ese Despacho, se logran superar dichas objeciones, dado que se han redactado las nuevas reivindicaciones 1 a 29, en el formato de preámbulo y parte caracterizante separados por la expresión "caracterizado porque"; y porque adicionalmente, la forma en la que se encuentran redactadas dichas reivindicaciones es completamente clara para una persona medianamente versada en la materia, la cual puede entender en forma apropiada que la invención consiste de un método caracterizado por sus etapas o pasos, así como de un sistema que integra una serie de elementos o dispositivo todos ellos tangibles (reales), así como a los medios legibles por computador y las estructuras relacionadas.

En vista de lo anterior, respetuosamente se solicita a ese Despacho reconocer la claridad de las reivindicaciones, toda vez que la forma en que se encuentran redactadas, y en la que se han caracterizados los objetos reclamados permiten entender en forma clara cual es el principio y el objeto de la invención, sin llevar a una mala interpretación de los mismos por parte de una persona medianamente versada en la técnica, además, de resultar concisas y completamente

sustentadas en la descripción presentada, tal y como es exigido por el artículo 30 de la Decisión 486.

- (3) De otra parte, me permito precisar que las reivindicaciones 1 a 29, si cumplen con los requisitos del citado artículo 14 de la D486, toda vez que definen claramente un método, un sistema, medios legibles por computador y estructuras relacionadas, todas y cada una de ellas mediante características técnicas, tal y como es requerido por la legislación Andina vigente; como se puede apreciar, por ejemplo, en la reivindicación 1, en la cual se caracteriza el método de la siguiente manera:

*"1. En un ambiente de computación, un método **CARACTERIZADO porque** comprende:*

en un primer dispositivo de computación, descubrir un segundo dispositivo de computación;

en el primer dispositivo, obtener datos relacionados con el segundo dispositivo, los datos que incluyen información de recursos del segundo dispositivo son formateados de acuerdo con un esquema entendido por el primer dispositivo; y el primer dispositivo interpreta los datos para acceder a un recurso del segundo dispositivo."(Subrayado fuera de texto).

Dicha caracterización claramente indica que el método reclamado, incluye pasos específicos, tales como desde un primer dispositivo descubrir un segundo dispositivo, mediante la obtención de datos que sean relacionados con dicho segundo dispositivo, además de que dichos datos son formateados de acuerdo con un esquema del primer dispositivo y en donde el primer dispositivo interpreta dichos datos para acceder a un recurso del segundo dispositivo. Los pasos enunciados corresponden a acciones específicas que se realizan o ejercen sobre elementos físicos, tales como el primer y segundo dispositivos, lo cual a todas luces cumple las exigencias establecidas en la legislación Andina vigente.

En efecto, una persona medianamente versada en la materia es conocedora de que la manipulación realizada de los dispositivos mencionados en la invención, que hacen parte fundamental del método y del sistema, no debe llegar a ser entendida como una simple secuencia de procesamiento de datos, sino que se debe entender como una secuencia de etapas que conlleva la manipulación de dispositivos específicos para llegar a obtener un resultados deseado, lo cual a todas luces corresponde a lo que se considera como un método y un sistema patentables.

De otra parte, ese Despacho insiste en negar la patentabilidad de la solicitud Colombiana, afirmando que las reivindicaciones presentan como medios de un sistema una secuencia de procesamiento de datos ejecutadas por computador. En este punto, es necesario llamar la atención de ese despacho al hecho de que tanto el método como los sistemas reivindicados, de ninguna manera caracterizan una secuencia de procesamiento de datos, como en forma equivocada ha sido interpretado por parte de ese despacho, toda vez que, el método y los sistemas reclamados se caracterizan por

acciones que se llevan a cabo sobre componentes físicos y reales que de ninguna forma pueden ser considerados como intangibles por parte de una persona medianamente versada en la técnica. En este sentido, es claro que el señor Examinador debe dirigir su análisis de claridad hacia los componentes físicos de los sistemas y las etapas del método, reivindicados, para no incurrir en el error de reducir los aspectos novedosos de la invención a un simple procesamiento de datos por computador.

- (4) De hecho, sobre este particular, y con respecto a la categoría de las reivindicaciones, el Manual Andino de Patentes establece en su numeral **4.4.2 Procedimiento**, lo siguiente:

"La segunda clase básica de reivindicaciones se aplica a todo tipo de actividades que supongan la utilización de cualquier producto material para efectuar el procedimiento. Dichas actividades se pueden ejercer sobre productos materiales, energía, otros procesos (por ejemplo procesos de control) o sobre cosas vivientes, cuando la legislación lo permite."

*Es esencial que la reivindicación estudiada no deje ninguna duda en cuanto a la categoría a la que pertenece."*¹.

En este sentido, resulta evidente que las reivindicaciones de la presente invención, están de acuerdo con las exigencias del citado artículo 14 de la D486, toda vez que se encuentran dirigidas a proteger procedimientos que no dejan ninguna duda de su naturaleza debido a que se definen en términos de etapas (Reivindicaciones 1 a 6), en donde las acciones unitarias de cada etapa son realizadas sobre productos materiales, tales como los dispositivos físicos referidos en la reivindicación 1, y productos definidos en términos de componentes (Reivindicaciones 7 a 29), como así se requiere en la práctica del ámbito de la propiedad Industrial.

Más aún, las reivindicaciones de método no solo pueden hacer referencia a los elementos que llevan a cabo dichas etapas, sino que deben hacerlo, pues de acuerdo con las enseñanzas del anterior aparte citado del Manual para la Evaluación de Patentes de la CAN, se tiene que **un método no puede estar limitado a una simple enumeración de las etapas** sin hacer mención de los componentes que están involucrados en tales etapas, lo correcto es que se indique en forma clara cuales son los elementos y las condiciones específicas a las cuales se debe llevar a cabo dicho método, pues de lo contrario, no podría ser implementado por una persona con conocimientos medios en la materia que estuviera interesada en desarrollar dicho proceso o método.

De hecho, si los método o procesos no hicieran referencia a los elementos y condiciones a las cuales se debe realizar, no sería posible reivindicar un proceso para la preparación de una sustancia X, pues al no indicar que componentes se mezclan o se procesan, y a

¹ Secretaría General de la Comunidad Andina. "Manual para el Examen de Solicitudes de Patentes de Invención en las Oficina de Propiedad Industrial de los Países de la Comunidad Andina". Quito., Ecuador., Junio de 2004. página 37.

que condiciones se hace, la persona que desee reproducir tal proceso, se encontraría completamente impedida para hacerlo, toda vez que una caracterización como la sugerida por ese Despacho, que no indique los elementos involucrados en el proceso, sería inadecuada y completamente confusa para la persona versada en la materia. La misma situación aplica para las reivindicaciones de método de la presente invención, porque es necesario establecer que elementos son los que intervienen en cada una de dichas etapas, para lograr el resultado deseado.

De la misma manera, los productos reivindicados están perfectamente definidos por componentes físicos estructurales que constituyen las piezas fundamentales para lograr el resultado propuesto por el inventor.

- (5) De otra parte, ese Despacho, nuevamente hace referencia al artículo 15 de la Decisión 486, con respecto a lo que no se considerarán invenciones, dentro de las que cita los programas de ordenadores o el soporte lógico como tales. Sobre este particular, ese Despacho establece que *"Al estudiar la solicitud, es evidente que tanto la descripción como las reivindicaciones que se refieren a **"en un ambiente de computación un método"**, **"un medio que se puede leer por computador que tiene instrucciones ejecutables por computador"**, **"en un ambiente de computación un sistema"**, **"un medio que se puede leer por computador que tiene almacenado en el mismo una estructura de datos"**, relacionan unas secuencias lógicas de procesamiento de datos (información), llevadas a cabo mediante el uso de medios técnicos como, por ejemplo, el ambiente informático de implementación convencional para exponer recursos como servicios a los que se accede con programas de software, el cual, se ilustra en las figuras 1A y 1B, junto con la descripción correspondiente. Dichos medios permiten procesar la información (las señales) del modo deseado, por ejemplo, para lograr el resultado relacionado en las reivindicaciones; por esta razón, el pliego es soporte lógico como tal, que no se considera una invención de conformidad con lo dispuesto en el citado artículo 15 e)."*

En este punto, si se analiza la invención dentro del contexto global se puede apreciar que **las reivindicaciones de la presente invención NO corresponden de ninguna manera a soporte lógico como tal**. En efecto, se ha demostrado que la presente invención reivindica un método definido en términos de etapas y un producto definido en términos de componentes estructurales, esto es, un sistema completo de elementos físicos que son manipulados en una forma específica para obtener el resultado deseado.

Por lo tanto, esta definición en términos de componentes estructurales no puede ser considerada como un soporte lógico como tal. En efecto, para una mejor comprensión de lo diferente que resulta la presente invención a un software o soporte lógico como tal, podemos recurrir a los conceptos básicos de lo que se debe entender como software (o soporte lógico), frente a lo reclamado en la invención en cuestión. El soporte lógico como tal, es un término genérico que **se aplica a los componentes NO físicos de un sistema informático**, como por

ejemplo, los programas de computador, sistemas operativos, etc., que permiten a esté, ejecutar sus tareas, tal y como se aprecia en la definición de software del Diccionario de la Real Academia de la Lengua Española.

software. (Voz inglesa). *1. m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.*

Teniendo en cuenta la clara definición de lo que es un software o soporte lógico como tal, resulta fácil visualizar las claras diferencias técnicas existentes entre el método, el sistema y los medios legibles por computador de la invención denegada, y un software, toda vez que en el presente caso los elementos manipulados son completamente tangibles (físicos), mientras el soporte lógico opera exclusivamente sobre intangibles, por lo tanto, no existe dentro de la materia objeto reclamada en las reivindicaciones 1 a 29 de la solicitud Colombiana en cuestión, una revelación explícita de un soporte lógico como tal.

En este sentido, es claro que la lectura de las reivindicaciones demuestra que los componentes que constituyen el método, el sistema, los medios legibles por computador, reivindicados son físicos, no abstractos, como si corresponde a un soporte lógico. De la misma manera, las etapas del proceso reivindicado son etapas válidas que conducen a un resultado deseado, lo cual no implica que se está reivindicando un soporte lógico como tal.

De hecho, si el resultado final logrado con estos objetos se implementa en un computador esto no es relevante para desvirtuar la patentabilidad de la materia objeto reivindicada. Más aún, resulta improcedente considerar las reivindicaciones como un soporte lógico, cuando las mismas están claramente definidas en términos de componentes y etapas, como así lo requiere la práctica en el campo de la propiedad intelectual.

En consecuencia, **las reivindicaciones de la presente invención NO corresponden a un soporte lógico como tal**, sino a productos definidos en términos de componentes físicos estructurales; y métodos perfectamente definidos en términos de una secuencia de etapas tendientes a lograr un resultado deseado.

- (6) Finalmente, en la Resolución Impugnada ese Despacho, señala que *"Por otra parte, se desconoce, además, lo dispuesto en el artículo 21 de la Decisión 486, porque se hace otro (segundo) uso de dispositivos convencionales o de partes de ellos, que en este caso consiste en el uso de un ambiente de computo, además, porque de forma explícita se describe y reivindica que el objeto se realiza en un ambiente de computación."*

Sobre este particular, y como ya se ha discutido ampliamente a lo largo del presente recurso, las reivindicaciones de la presente invención se refieren explícitamente a un método, un sistema, y medios tangibles (físicos), que de ninguna manera corresponden a un

segundo uso, o a un uso diferente de una invención ya conocida, como lo sugiere ese Despacho en la Resolución impugnada.

De hecho, el artículo 21 de la Decisión 486, señala: "**Los productos o procedimientos ya patentados, comprendidos en el estado de la técnica, de conformidad con el artículo 16 de la presente decisión, no serán objeto de nueva patente, por el simple hecho de atribuirse un uso distinto al originalmente comprendido por la patente inicial**" (Subrayado fuera de texto).

Del texto del artículo 21, se colige que la prohibición de patentabilidad hace referencia **a productos o procedimientos que ya gozan de patente**, pero no se refiere a cada uno de los componentes que formen parte de dicho producto o procedimiento. En efecto, el real alcance del artículo 21 de la Decisión 486, se da para solicitudes de patente en las cuales, se reclama un uso diferente al originalmente reivindicado, para exactamente la misma materia objeto ya protegida por patente. Por el contrario, en el presente caso no se incurre en esta situación, toda vez que los elementos que hacen parte de la invención cumplen exactamente la misma función para que fueron hechos originalmente. En este punto, lo realmente importante, es que se debe evaluar la patentabilidad del conjunto total de la invención, y no de cada uno de sus elementos por separado.

Por consiguiente, la aplicación del artículo 21 de la Decisión 486 sobre los elementos que conforman la materia objeto de la presente invención es completamente improcedente, porque el conjunto y la relación técnica entre los mismos, que hacen posible obtener el resultado deseado, no corresponden a un segundo uso de los elementos que conforman la invención. De hecho, la invención se debe analizar dentro del contexto como "un todo", y no por cada uno de sus componentes, como ya se ha manifestado anteriormente.

- (7) Ahora bien, con el fin de mostrar a ese Despacho que los argumentos presentados a lo largo del presente recurso, así como los argumentos presentados anteriormente como respuesta al citado oficio No. 4276, no son simples aseveraciones del solicitante que carecen del debido sustento técnico y legal, respetuosamente me permito de acuerdo con el artículo 52 del Código Contencioso Administrativo, el cual en su numeral 3, establece que en los Recursos se deberán relacionar las pruebas que se pretende hacer valer; presentar como pruebas de la patentabilidad de la presente solicitud, dos documentos de patente que cobijan exactamente la misma materia objeto de la solicitud Colombiana en cuestión, y que corresponde a la patente Mexicana **No. 240176** y la patente Europea **EP 1376977 B1**.

En este punto, atentamente me permito reiterar el hecho de que la misma materia objeto de la solicitud Colombiana **No. 03-046.838**, ha sido evaluada por parte de Examinadores técnicos y legales, de las Oficinas de patentes de México (IMPI) y Europa (EPO), los cuales, debido a la amplia y conocida homogenización del conocimiento que se debe llevar por parte de todas las oficinas nacionales de propiedad intelectual, deben contar con la misma formación profesional e idoneidad de los señores Examinadores de la Oficina de Nuevas

335

Creaciones de la SIC, para llevar a cabo el correcto análisis de patentabilidad de la materia objeto reivindicada por el solicitante de la misma invención denegada en Colombia.

Vale la pena llamar la atención de ese Despacho al hecho de que como resultado del examen de patentabilidad realizado en cada una de dichas oficinas, los Examinadores encontraron en forma independiente, que la materia objeto reclamada en las mismas reivindicaciones 1 a 29 de la solicitud en cuestión, cumplen a cabalidad con los requisitos técnicos y legales exigidos por la legislación vigente en cada una de dichas oficinas, y en vista de esto, han concedido el privilegio de patente para dichas solicitudes, bajo los correspondientes **Nos. 240176 y EP 1376977 B1**.

El hecho de que los Examinadores de dos oficinas distintas, hayan encontrado los suficientes meritos para conceder el privilegio de patente para la misma materia objeto de la solicitud colombiana, que se encuentra en cuestión, debe claramente constituir un fuerte indicio que permite demostrar que la presente solicitud sí cumple con los requisitos necesarios de claridad, novedad, nivel inventivo y aplicación industrial, requeridos sobre las enseñanzas del estado del arte anterior, como para que le sea concedido el privilegio de patente solicitado.

Ahora bien, frente al hecho de la concesión de las dos patentes citadas anteriormente, podrían surgir algunos interrogantes, como los siguientes:

- ¿Por qué dos oficinas diferentes conceden una patente para la misma materia objeto de la solicitud Colombiana No. 03-046.838?, y
- ¿Por qué la concesión de dichas patentes debe ser considerada como un fuerte indicio de patentabilidad de dicha materia objeto, por parte de otra oficina de patentes, específicamente por parte de la Oficina de Nuevas Creaciones de la SIC?

Para dar respuesta a estos dos interrogantes, se debe recurrir inevitablemente a la legislación vigente en materia de patentes, en cada uno de los países que ya ha concedido dichas patentes, y a la legislación Andina, para poder establecer si se ciñen a los mismos principios legales y técnicos para la evaluación de la patentabilidad.

En conexión con lo anterior, se presentan en las columnas dos, tres y cuatro del siguiente cuadro comparativo, los artículos propios de cada una de dichas legislaciones, referentes a los requisitos de patentabilidad exigidos, a lo que no se considera patentable, al requisito de claridad de las reivindicaciones y al uso o segundo uso, toda vez, que son dichos artículos los que motivan la negación emitida por parte de ese Despacho.

Cuadro Comparativo:

	Normativa Andina (Decisión 486) ²	Ley de propiedad Intelectual de México ³	Convención Europea de Patentes (EPC) ⁴
--	---	--	--

² www.emercatoria.edu.co/PAGINAS/volumen3/Documentos01/DECISION%20486.doc

336

REQUISITOS DE PATENTABILIDAD	Art 14.- Los Países Miembros otorgarán patentes para las invenciones, sean de producto o de procedimiento, en todos los campos de la tecnología, <u>siempre que sean nuevas, tengan nivel inventivo y sean susceptibles de aplicación industrial.</u>	Art 16.- Serán patentables las invenciones <u>que sean nuevas, resultado de una actividad inventiva y susceptibles de aplicación industrial</u>, en los términos de esta Ley, excepto: I.- Los procesos esencialmente biológicos para la producción, reproducción y propagación de plantas y animales; II.- El material biológico y genético tal como se encuentran en la naturaleza; III.- Las razas animales; IV.- El cuerpo humano y las partes vivas que lo componen, y V.- Las variedades vegetales.	Art. 52 Invenciones Patentables (1) Las patentes europeas <u>serán concedidas para cualquier invención, en todos lo campos de la tecnología, con la condición de que sean nuevas, involucren un nivel inventivo y sean susceptibles de aplicación industrial.</u> (2) Las siguientes en particular no serán consideradas como invenciones dentro del significado del párrafo 1: (a) descubrimientos, teorías científicas y métodos matemáticos; (b) creaciones estéticas; (c) esquemas, reglas y métodos para realizar actos mentales, jugar juegos o hacer negocios, y programas para computadores; (d) presentaciones de información.
NO SE CONSIDERAN INVENCIONES	Art 15.- No se considerarán invenciones: a) los descubrimientos, las teorías científicas y los métodos matemáticos; b) el todo o parte de seres vivos tal como se encuentran en la naturaleza, los procesos biológicos naturales, el material biológico existente en la naturaleza o aquel que pueda ser aislado, inclusive genoma o germoplasma de cualquier ser vivo natural; c) las obras literarias y artísticas o cualquier	Art 19.- No se considerarán invenciones para los efectos de esta Ley: I.- Los principios teóricos o científicos; II.- Los descubrimientos que consistan en dar a conocer o revelar algo que ya existía en la naturaleza, aún cuando anteriormente fuese desconocido para el hombre; III.- Los esquemas, planes, reglas y métodos para realizar actos mentales, juegos o negocios y los métodos matemáticos; IV.- Los programas de computación;	Art. 52 Invenciones Patentables (2) <u>Las siguientes en particular no serán consideradas como invenciones</u> dentro del significado del párrafo 1: (a) descubrimientos, teorías científicas y métodos matemáticos; (b) creaciones estéticas; (c) esquemas, reglas y métodos para realizar actos mentales, jugar juegos o hacer negocios, y programas para computadores;

³http://www.impi.gob.mx/index.php?option=com_content&task=view&id=98&Itemid=206&limit=1&limitstart=0
⁴[http://documents.epo.org/projects/babylon/eponet.nsf/0/E4F8409B2A99862FC125736B00374CEC/\\$File/EPC_13th_edition.pdf](http://documents.epo.org/projects/babylon/eponet.nsf/0/E4F8409B2A99862FC125736B00374CEC/$File/EPC_13th_edition.pdf)

	otra protegida por el derecho de autor; d) los planes, reglas y métodos para el ejercicio de actividades intelectuales, juegos o actividades económico-comerciales; <u>e) los programas de ordenadores o el soporte lógico, como tales;</u> y, f) las formas de presentar información.	V.- Las formas de presentación de información; VI.- Las creaciones estéticas y las obras artísticas o literarias; VII.- Los métodos de tratamiento quirúrgico, terapéutico o de diagnóstico aplicables al cuerpo humano y los relativos a animales, y VIII.- La yuxtaposición de invenciones conocidas o mezclas de productos conocidos, su variación de uso, de forma, de dimensiones o de materiales, salvo que en realidad se trate de su combinación o fusión de tal manera que no puedan funcionar separadamente o que las cualidades o funciones características de las mismas sean modificadas para obtener un resultado industrial o un uso no obvio para un técnico en la materia.	(d) presentaciones de información.
CLARIDAD DE LAS REIVINDICACIONES	Art 30.- Las reivindicaciones definirán la materia que se desea proteger mediante la patente. <u>Deben ser claras y concisas y estar enteramente sustentadas por la descripción.</u> Las reivindicaciones podrán ser independientes o dependientes. Una reivindicación será independiente cuando defina la materia que se desea proteger sin referencia a otra reivindicación anterior. Una reivindicación será dependiente cuando defina la materia que se desea proteger refiriéndose a una reivindicación anterior. Una reivindicación que se refiera a dos o más reivindicaciones	Art 47.- A la solicitud de patente se deberá acompañar: I.- La descripción de la invención, [...] II.- Los dibujos que se requieran para la comprensión de la descripción; III.- <u>Una o más reivindicaciones, las cuales deberán ser claras y concisas y no podrán exceder del contenido de la descripción,</u> y IV.- Un resumen de la descripción de la invención, que servirá únicamente para su publicación y como elemento de información técnica.	Art. 84 Reivindicaciones Las reivindicaciones definirán la materia para la cual se busca protección. <u>Deben ser claras y concisas y estar sustentadas por la descripción.</u>

	anteriores se considerará una reivindicación dependiente múltiple.		
USOS	Art 21.- Los productos o procedimientos ya patentados, comprendidos en el estado de la técnica, de conformidad con el artículo 16 de la presente Decisión, <u>no serán objeto de nueva patente, por el simple hecho de atribuirse un uso distinto al originalmente comprendido por la patente inicial.</u>	Art 19.- No se considerarán invenciones para los efectos de esta Ley: I.- Los principios teóricos o científicos; II.- Los descubrimientos que consistan en dar a conocer o revelar algo que ya existía en la naturaleza, aún cuando anteriormente fuese desconocido para el hombre; III.- Los esquemas, planes, reglas y métodos para realizar actos mentales, juegos o negocios y los métodos matemáticos; IV.- Los programas de computación; V.- Las formas de presentación de información; VI.- Las creaciones estéticas y las obras artísticas o literarias; VII.- Los métodos de tratamiento quirúrgico, terapéutico o de diagnóstico aplicables al cuerpo humano y los relativos a animales, y <u>VIII.- La yuxtaposición de invenciones conocidas o mezclas de productos conocidos, su variación de uso, de forma, de dimensiones o de materiales,</u> salvo que en realidad se trate de su combinación o fusión de tal manera que no puedan funcionar separadamente o que las cualidades o funciones características de las mismas sean modificadas para obtener un resultado industrial o un uso no obvio para un técnico en la materia.	En la EPC solo se aceptan los segundos usos de sustancias con propiedades de medicamentos (Artículo 53 (c)), y su forma de ser presentados es conocida como “ Suiss type ”.

El cuadro anterior, casi por si solo da respuesta a los interrogantes planteados, toda vez que al hacer una simple comparación de cada uno de los requisitos exigidos por las diferentes legislaciones, se puede ver que son los mismos requisitos tanto legales como técnicos, pues por ejemplo, si tomamos el artículo 14 de la Decisión 486, el cual establece que se otorgaran patentes a las invenciones, siempre y cuando sean novedosas, tengan nivel inventivo y sean susceptibles de aplicación industrial, y lo comparamos con el artículo 16 de la ley de Patentes Mexicana, y con el Artículo 52 de la Convención Europea de Patentes (EPC), encontramos que son exactamente los mismos requisitos técnicos y legales los exigidos en México, Europa y Colombia, lo mismo ocurre con los artículos 30 de la D486, 47 de la Ley de Patentes Mexicana y 84 de la EPC, que exigen en forma idéntica, que las reivindicaciones cumplan con los requisitos de claridad, concisión y que sean debidamente sustentadas por la descripción; esta misma situación se presenta al comparar cualquier otro de los artículos de la legislación Andina, enunciados en el cuadro comparativo anterior, con sus artículos equivalentes en las legislaciones Mexicana y Europea.

Esto demuestra en forma evidente, que si los examinadores técnicos y legales de dos oficinas de propiedad industrial e intelectual como la IMPI y la EPO, han llegado a la conclusión de que el método, los medios legibles por computador, el sistema y las estructuras reclamadas en las actuales reivindicaciones 1 a 29 de las respectivas patentes concedidas por dichas oficinas, son claras y patentables a la luz de los requisitos exigidos y de las excepciones de patentabilidad existentes en sus legislaciones, dicho método, medios legibles por computador, sistema y estructuras reclamadas en las actuales reivindicaciones 1 a 29 de la solicitud Colombiana en cuestión, también podrían llegar a ser patentables bajo la legislación Andina vigente, debido a las sorprendentes similitudes entre los artículos 14, 15 e), 21 y 30 de la Decisión 486, y los artículos 16, 19 IV), 47 y 19 VIII), de la Ley de Patentes de México, y los artículos 52 (1), 52 (2), 84 de la EPC.

Entonces, para entender porque dos oficinas diferentes conceden una patente para la misma materia objeto de la solicitud Colombiana No. 03-046.838, no es necesario sino comprender que esto se debe a que la materia objeto de la solicitud Colombiana, resulta clara para una persona medianamente versada en la materia, y además, no es un programa de computador o un soporte lógico como tal, y mucho menos corresponde a un uso diferente de elementos ya conocidos, pues expertos con los mismos conocimientos técnicos y bajo los mismos requisitos legales que son aplicados por los expertos Colombianos de ese Despacho, han podido encontrar dentro de la materia objeto reclamada, es decir, el método, el sistema, los medios legibles por computador, y las estructuras de las reivindicaciones 1 a 29, los suficientes elementos técnicos que le brindan novedad, altura inventiva y aplicación industrial, a una invención que representa un avance en el estado actual de la técnica.

En vista de las anteriores comparaciones, resulta fácil entender porque la concesión de dichas patentes por parte de la IMPI y la EPO,



debe ser considerada como un fuerte indicio de patentabilidad de la materia objeto de la presente invención, que no debería ser desestimado o pasado por alto por parte de la Oficina de Nuevas Creaciones de la SIC, toda vez que, los conceptos de otras oficinas pueden ser empleados como una muy útil herramienta en la evaluación de la patentabilidad de una solicitud en particular. Esta apreciación, no corresponde a una simple aseveración hecha por el solicitante, sino que es una consideración que tiene su justificación en la misma normativa Andina en materia de patentes. En efecto, el empleo de documentos y estudios provenientes de otras oficinas de patentes, es una práctica reconocida y aceptada por el Artículo 46 de la Decisión 486, el cual en algunos de sus apartes establece lo siguiente:

"Artículo 46.- La oficina nacional competente podrá requerir el informe de expertos o de organismos científicos o tecnológicos que se consideren idóneos, para que emitan opinión sobre la patentabilidad de la invención. Asimismo, cuando lo estime conveniente, podrá requerir informes de otras oficinas de propiedad industrial.

[...]

- a) copia de la solicitud extranjera;
- b) copia de los resultados de exámenes de novedad o de patentabilidad efectuados respecto a esa solicitud extranjera;
- c) copia de la patente u otro título de protección que se hubiese concedido con base en esa solicitud extranjera;
- d) copia de cualquier resolución o fallo por el cual se hubiese rechazado o denegado la solicitud extranjera; o,
- e) copia de cualquier resolución o fallo por el cual se hubiese anulado o invalidado la patente u otro título de protección concedido con base en la solicitud extranjera.

La oficina nacional competente podrá reconocer los resultados de los exámenes referidos en el literal b) como suficientes para acreditar el cumplimiento de las condiciones de patentabilidad de la invención. [...]". (Subrayado fuera de texto).

El artículo 46, claramente faculta a los examinadores técnicos y legales a considerar como elementos importantes para determinar la patentabilidad de una invención, las opiniones y los resultados de la examinación realizada por parte de los funcionarios de otras oficinas de patentes.

Ahora bien, el solicitante claramente entiende que el hecho de que la misma materia objeto de la solicitud Colombiana en cuestión, haya sido concedida tanto por parte de la oficina Mexicana (IMPI), como por parte de la EPO, no obliga a la concesión de dicha patente por parte otra oficina, específicamente, por parte de la Oficina de Nuevas

Creaciones de la SiC, pero lo cierto es que como ya se ha discutido ampliamente, **desde el punto de vista técnico y legal, dicha concesión SI constituye un fuerte indicio que resulta mas que razonable de que la materia objeto de la solicitud en cuestión, posee efectivamente los requisitos de claridad, novedad, nivel inventivo y aplicación industrial necesarios para gozar del privilegio de patente solicitado**, debido a que una vez analizada por personas expertas en la materia, con la misma formación profesional e idoneidad del señor Examinador Colombiano, y bajo los mismos parámetros de patentabilidad, dichas personas han encontrado los elementos necesarios para considerar que la materia objeto ahora reivindicada, es clara, novedosa, inventiva y aplicable industrialmente.

CONCLUSION

En primer lugar, las modificaciones realizadas en el capítulo reivindicatorio como respuesta al anterior concepto técnico emitido por ese Despacho, permiten que las reivindicaciones 1 a 29 de la presente solicitud cumplan a cabalidad con las exigencias del Artículo 30 de la Decisión 486.

De otra parte, los argumentos expuestos al lo largo del presente escrito, que se encuentran perfectamente sustentados, incluso por el hecho de que la misma materia reclamada en la invención denegada ha sido reconocida como patentable por parte de los examinadores de dos oficinas de patentes, como la Oficina Mexicana (IMPI) y la Oficina europea (EPO), ponen de manifiesto de una manera clara y contundente, que la materia objeto de la presente invención no se encuentra incluida en lo que se considera como materia no patentable bajo los parámetros legales y técnicos de la Decisión 486, específicamente en el Artículo 15 e); y que tampoco corresponde a un uso diferente de elementos conocidos por lo que no esta cubierta por lo establecido en el artículo 21 de la misma Decisión 486.

Por lo tanto, la invención contenida en la solicitud Colombiana No. 03-046.838, resulta completamente clara, concisa, correctamente caracterizada en forma técnica en procedimientos y productos, los cuales resultan ser patentables toda vez que son materia considera como patentable que reviste Novedad, Altura Inventiva y Aplicación Industrial requeridas por la Normativa Andina vigente en Colombia, con características técnicas perfectamente definidas que permiten obtener un resultado deseado que ofrece una solución a un problema aún existente en el estado del arte, lo cual hace que la invención sea un verdadero aporte novedoso al estado del arte.

2. En consecuencia, atentamente me permito solicitar a ese Despacho, se sirva **tener en cuenta los argumentos aquí expuestos** y las pruebas aportadas, y con base en ellos, revocar la decisión contenida en la Resolución impugnada No. **11341** para llevar nuevamente a cabo el estudio de patentabilidad de la presente solicitud y conceder el privilegio de patente solicitado para la materia objeto de las reivindicaciones 1 a 29 de la presente invención, toda vez que **ya se ha reconocido por algunos expertos en la materia, que la misma cumple con los mismos**

requisitos de claridad, novedad, nivel inventivo y aplicación industrial que exige la normativa andina en materia de patentes.

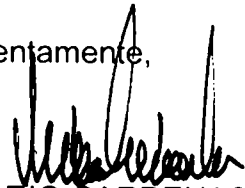
3. Pruebas

- Copia informal del documento "PROTECCIÓN POR PATENTES DE INVENCIONER RELACIONADAS CON PROGRAMAS DE ORDENADOR" AIPPI, Enero 2, 2006.
- Copia informal del expediente de la patente Mexicana No. 240176, y
- Copia informal del expediente de la patente Europea No EP 1376977 B1, que cobijan la misma materia objeto de la solicitud Colombiana No. 03-046.838.

4. NOTIFICACIONES

El suscrito(a) como apoderado(a) de la sociedad **MICROSOFT CORPORATION**, recibirá notificaciones en la Secretaría de ese Despacho o en la Carrera 7ª No. 71-52 Torre B Piso 9, de la ciudad de Bogotá D.C.

Atentamente,

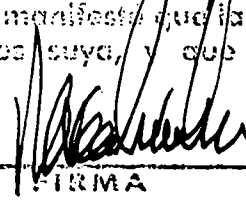

DARIO CARDENAS NAVAS
c.c. 17.066.629 de Bogotá
t.p. 1.250 C.S.J.

P. 216

COMPARECENCIA PERSONAL Y AUTENTICACION DE FIRMA

La Notaria Dieciséis (16) de Bogotá, da fé que el anterior escrito dirigido a: Beatriz Vargas Navarro fue presentado personalmente por: Dario Cardenas Navas

quien exhibió la C.C. No. 17066629 de Bogotá D.C. No. 1250 y manifestó que la firma que aparece en el presente documento es suya, y que acepta el contenido del mismo.


FIRMA

BEATRIZ VARGAS NAVARRO
NOTARIA



2 de enero del 2006



PROTECCIÓN POR PATENTES DE INVENCIONES RELACIONADAS CON PROGRAMAS DE ORDENADOR

Este documento ha sido preparado por el Comité Especial de AIPPI/Q132 con la participación de:

- por Europa:
 - Michel de Beaumont (Francia),
 - Stephan Freischerm (Alemania),
 - Fernand de Visscher (Bélgica),
 - Jürgen Betten (Alemania),

- por Estados Unidos:

- Erwin Basinski,
- Ronald S. Laute,

- por Japón:

- Yoshikazu Tani,
- Mishihiro Miyasaka,

- otros:

- José Antonio Fana Correa (Brasil),

y con la ayuda del equipo del Relator General de la AIPPI.

...

El propósito de este documento es dar una visión general de la situación actual relativa a la patentabilidad de invenciones relacionadas por ordenador... o relacionadas con programas de ordenador... y revisar algunas de las ventajas económicas de la protección de las patentes.

Después de unos comentarios generales, este documento:

- resumirá la situación en los Estados Unidos, Japón y Europa,
- comparará estas situaciones y recordará la posición de AIPPI,
- presentará algunos argumentos económicos.

1. COMENTARIOS GENERALES

La patentabilidad de las innovaciones relacionadas con programas de ordenador ha sido objeto de un vivo debate entre los usuarios de la propiedad intelectual y los expertos en tecnologías de la información durante los últimos 40 años. Esta discusión es muy similar a las discusiones anteriores en relación con el reconocimiento de la protección mediante patentes a nuevos e importantes sectores tecnológicos tal como la extensión a sustancias médicas (aproximadamente hace 100 años). Existe un debate similar en relación con la biotecnología.

Las economías de los países industrializados son de forma creciente dependientes del sector terciario (industria de servicios), que incluye el sector de los servicios financieros. Los nuevos desarrollos en los servicios empresariales son generalmente nuevos métodos de trabajo implementados muy frecuentemente a través del uso de redes de ordenadores tales como Internet con la ayuda de ordenadores. La cuestión de la existencia de la protección por patente para los programas de ordenador o a los métodos comerciales pasa a ser, por tanto, una cuestión de aplicar el sistema de protección conocido al sector económico con un crecimiento más fuerte.

Inicialmente, no se otorgaba protección por patente a las invenciones relacionadas con programas de ordenador en la mayoría de los sistemas de patentes pero, de forma gradual, el alcance de la protección por patentes se ha ido extendiendo. El intento de la Unión Europea de clarificar y armonizar los estándares de protección por patente de las invenciones relacionadas con programas de ordenador falló después de una disputa muy agitada sobre la extensión de la protección por patente que debía otorgarse a tales invenciones y los peligros y beneficios resultantes de la protección por patente. En la India las propuestas para examinar el régimen de patentes aplicable a las invenciones relacionadas con programas de ordenador sufrieron una suerte similar. Esta disputa se intensificó con el desarrollo de la comunidad del software de código abierto/libre. La subiguiente introducción de métodos comerciales implementados por ordenador parece haber magnificado la disputa a medida que bancos y otras industrias de servicios financieros pasaron a preocuparse por estas patentes en los Estados Unidos.

Las invenciones relacionadas con programas de ordenador o las invenciones implementadas por ordenador son invenciones cuya implementación implica la utilización de un ordenador, una red de ordenadores u otros aparatos programables, estando al menos una de las características de la invención realizada a través de un programa de ordenador. En la mayoría de los casos tales invenciones están dirigidas a una nueva funcionalidad que debe ser ejecutada por medio de un ordenador u otro dispositivo programable.

Las invenciones relacionadas con programas de ordenador penetran en casi todos los ámbitos de la tecnología. Algunos ejemplos de tales invenciones son:

- ... métodos de transmisión de datos que aceleran la comunicación inalámbrica;
- ... métodos para la encriptación de datos que hacen la comunicación de datos más segura;
- ... control en la motorización de vehículos, tales como los modernos métodos de inyección de fuel o el control de un sistema de motorización híbrido;
- ... detección de condiciones peligrosas cuando se conduce un automóvil y reacción automática a tales condiciones;
- ... aplicaciones GPS que combinan datos de posición con datos de viaje y datos de un mapa para esquivar el tráfico y facilitar instrucciones de viaje que se adecuen a las necesidades del conductor;

343

..... métodos promocionales sobre redes de datos digitales y métodos para optimizar información de negocios e información financiera utilizando las fuentes disponibles en Internet;

..... máquinas-cajero automáticas para dispensar dinero en efectivo en lugares alrededor del mundo.

La materia patentable en tales innovaciones son los nuevos aparatos con los programas y/o el método integrados, es decir, el nuevo mecanismo para alcanzar un fin específico, práctico, concreto y útil y no la implementación de un programa de ordenador específico. El código fuente del programa de ordenador se protege por la propiedad intelectual (derecho de autor o copyright) y no por las patentes. La protección a través de la propiedad intelectual (derecho de autor/copyright) únicamente protege la expresión específica del programa (por ejemplo, el listado o el código específico) contra las copias mientras que la protección por patente protege los elementos de un nuevo método (por ejemplo, la funcionalidad) independientemente de la implementación del código específico en un programa. La protección que otorga la ley de propiedad intelectual (derecho de autor/copyright) y la protección que otorga la ley de patentes, aunque se aplique al mismo producto de software, cubre aspectos completamente diferentes del producto y no deben confundirse. La propiedad intelectual (derecho de autor/copyright) otorga una protección fuerte contra la simple multiplicación (por ejemplo, la copia directa o el uso no autorizado) de un producto de software. La patente protege contra la copia del aparato y los métodos que puedan utilizar un programa de ordenador. Además la propiedad intelectual (derecho de autor/copyright) y los derechos de patente son de una naturaleza diferente. En la práctica, la expresión de una idea es protegible por la propiedad intelectual (derecho de autor/copyright) por el mero hecho de que se trata de una creación, por ejemplo la expresión de cualquier programa de ordenador original es protegible por la propiedad intelectual (derecho de autor/copyright). Por el contrario, para que sea patentable, una invención debe ser nueva y no evidente (no obvia) y esta condición no es fácilmente cumplimentada por un producto de software. Por ejemplo, un programa de ordenador que meramente implementa una determinada especificación no es patentable si solamente se han utilizado para la implementación de técnicas de programación normales. Solamente pueden ser inventivas y definidas en una reivindicación de patente la materia nueva y no evidente de la especificación y/o implementaciones funcionales de la especificación nuevas y no evidentes.

En este documento, estudiaremos si las innovaciones implementadas por programas de ordenador son aptas para ser patentadas. Debe tenerse en cuenta que entre estas innovaciones aptas para ser patentadas, únicamente un pequeño número será patentable: aquellas que sean nuevas y no evidentes.

2. SITUACIÓN ACTUAL

2.1. ESTADOS UNIDOS DE AMÉRICA

En los Estados Unidos, las patentes de programas de ordenador y de métodos comerciales son objeto de las mismas reglas que se aplican a todas las invenciones. Una invención tiene derecho a ser protegida por patente si satisface los siguientes requisitos: utilidad (35 U.S.C. ••101), novedad, no obviedad, y descripción escrita clara

Los requisitos legales del artículo 35 U.S.C. ••101 son: "cualquier proceso, máquina, fabricación, o composición nuevos y útiles, o cualquier perfeccionamiento nuevo y útil de los mismos ...".

El texto de la norma legal sugiere una interpretación liberal y el Congreso de los Estados Unidos ha señalado que la Sección 101 debería incluir "cualquier cosa bajo el sol hecha por el hombre" (Diamond v. Chakrabarty). En efecto, los tribunales norteamericanos han reconocido que la Sección 101 debería aplicarse libremente a las nuevas tecnologías, con independencia de que el Congreso previera el avance tecnológico. Quizás esta es la razón por la cual los principios de la ley han permanecido virtualmente inalterados durante 200 años en los Estados Unidos.

Aunque los tribunales norteamericanos interpretan la Sección 101 de forma amplia, han reconocido tres excepciones, creadas judicialmente, a la materia que puede ser patentable: las leyes de la naturaleza, los fenómenos naturales, y las ideas abstractas. Esta última excepción, las ideas abstractas, ha planteado problemas históricos a las invenciones de programas de ordenador y de métodos comerciales.

La prueba que actualmente utiliza el Tribunal de Apelaciones del Circuito Federal de los Estados Unidos para determinar si existe materia patentable es si la reivindicación contiene un resultado útil, concreto y tangible, y al efectuar dicha prueba, el foco de atención se encuentra en la aplicación práctica que equivale a utilidad práctica. El Circuito Federal ha señalado que una idea abstracta por sí misma nunca satisface los requisitos del artículo 35 U.S.C. ••101. Sin embargo, una idea abstracta cuando se aplica de forma práctica para producir un resultado útil, concreto y tangible satisface la Sección 101 (State Street Bank & Trust Co. v. Signature Financial Group, Inc.).

Una revisión de los casos del Circuito Federal desde la decisión del Tribunal Supremo norteamericano en Diamond v. Diehr (450 U.S. 175 (1981)), incluyendo tanto los casos State Street como AT&T, permite percibir los tipos de invenciones que el Circuito Federal ha encontrado y continuará encontrando patentables. Estos casos revelan que las reivindicaciones que han descrito máquinas, manipulación de máquinas, o procesos entre máquinas se ha considerado que describen materia patentable. De forma similar, las reivindicaciones que describen una transformación física o la generación de datos se ha considerado que describen materia patentable. Por otro lado, las reivindicaciones que han descrito ideas abstractas por sí mismas (por ejemplo, como tales) y las reivindicaciones que han descrito únicamente etapas efectuadas por un ser humano se ha considerado que no describen materia patentable. Por tanto, los intentos de reivindicar ideas abstractas por sí mismas o métodos efectuados únicamente por un ser humano corren el riesgo de tener problemas de invalidez bajo la Sección 101.

Resumen

De acuerdo con la práctica norteamericana, cualquier invención útil hecha por el hombre es patentable, incluyendo cualquier programa de ordenador y método comercial nuevo y no evidente.

2.2. JAPÓN

En Japón, las invenciones relacionadas con programas de ordenador son patentables si satisfacen los requisitos de la Ley de Patentes japonesa que se aplican a otras invenciones, es decir, invenciones previstas por la ley (Secciones 2(1) y 29(1)), novedad, actividad inventiva, requisitos sobre descripción. En cuanto a las invenciones relacionadas con negocios que utilizan programas de ordenador, se examinan de la misma manera que las invenciones relacionadas con programas de ordenador.

Con el fin de analizar las únicas cuestiones de examen que presentan estos tipos de invenciones, la Oficina Japonesa de Patentes publicó unas Guías de Examen para invenciones relacionadas con programas de ordenador. Las Guías de Examen explican, con ejemplos específicos, qué tipos de invenciones relacionadas con programas de ordenador satisfacen los requisitos, incluyendo el concepto legal de invención y la actividad inventiva.

El concepto legal de invención está definido por la Sección 2(1) de la Ley de Patentes japonesa como «una creación (altamente avanzada) de ideas técnicas que utilizan una ley de la naturaleza».

Puesto que debe utilizarse una ley de la naturaleza, no todas las invenciones constituyen invenciones previstas por la ley. Por ejemplo, una ley de la naturaleza por sí misma, un fenómeno natural, reglas hechas por el hombre tales como las leyes de economía, esquemas/métodos de negocios, ideas abstractas, algoritmos puramente matemáticos, disposiciones arbitrarias, actividades mentales, la mera presentación de información, listados de programas de ordenador no constituyen invenciones previstas por la ley.

En cuanto a las invenciones relacionadas con programas de ordenador, de acuerdo con las Guías de Examen, a menos que se trate de una invención de este tipo no típica, tal como una invención que controle un aparato (por ejemplo, una lavadora, un motor, un driver de disco duro), la cuestión de si constituyen o no invenciones previstas por la ley se juzga analizando si el proceso de información por programa de ordenador se realiza concretamente utilizando recursos de hardware (por ejemplo, CPU, memoria).

En otras palabras, las invenciones relacionadas con programas de ordenador tienen que ser descritas en una reivindicación de tal manera que los recursos de software y hardware trabajen de forma cooperativa. La mera descripción de los recursos de hardware (tales como CPU, ROM) no es suficiente.

En cuanto a las invenciones relacionadas con programas de ordenador que incluyen etapas efectuadas por un ser humano, no constituyen usualmente una invención prevista por la ley, puesto que una invención en su conjunto ha de utilizar una ley de la naturaleza.

Si una invención relacionada con un programa de ordenador constituye una invención de acuerdo con la ley, es patentable en forma de un aparato, un método, un programa, o un medio de almacenamiento que almacena un programa que puede ser leído por un ordenador.

Sin embargo, la invención debe implicar una actividad inventiva. De acuerdo con las Guías de Examen, por ejemplo, usualmente no implican una actividad inventiva: 1) la aplicación del estado de la técnica a otros sectores (por ejemplo, extracción de información médica de extracción de información de usuario), 2) la implementación de funciones por software que son implementadas por hardware en el estado de la técnica, y 3) la sistematización de transacciones que eran efectuadas por seres humanos en el estado de la técnica.

Cuando se valora la actividad inventiva de una invención implementada por ordenador, la persona experta en la materia que debe determinar si existe actividad inventiva se considera que tiene un conocimiento del sector de aplicación del software (por ejemplo, sector financiero) y en el sector de la tecnología de los ordenadores. Ello parece implicar que la contribución inventiva puede también ser efectuada en el sector no técnico (por ejemplo, sector financiero) en la medida en que el producto

reivindicado satisfaga los criterios antes mencionados de invención previstos por la ley

Resumen

En Japón, las invenciones de programas de ordenador e incluso los métodos comerciales implementados por ordenador son patentables, si se define en las reivindicaciones una interacción específica con un recurso de hardware.

2.3. EUROPA

La base legal es el Artículo 52 del Convenio de la Patente Europea (CPE):

- (1) Las patentes europeas serán concedidas para las invenciones nuevas que supongan una actividad inventiva y que sean susceptibles de aplicación industrial.
- (2) No se considerarán invenciones a los efectos del párrafo 1 en particular:
 - a) los descubrimientos, las teorías científicas y los métodos matemáticos;
 - b) las creaciones estéticas;
 - c) los planes, principios y métodos para el ejercicio y actividades intelectuales, para juegos o actividades económicas, así como los programas de ordenadores;
 - d) las formas de presentar informaciones.

- 3) Lo dispuesto en el párrafo 2 excluye la patentabilidad de los elementos enumerados en el mismo solamente en la medida en que la solicitud de patente europea o la patente europea no se refiera más que a uno de estos elementos considerados como tales.

Ello implica que «los programas de ordenador» y «los métodos comerciales» están excluidos «como tales» de la protección por patente.

Sin embargo, es todavía objeto de discusión y no está claro qué significa, por ejemplo, un «programa de ordenador como tal». Probablemente es un programa de ordenador sin carácter técnico o un programa de ordenador que no resuelva un problema técnico.

Ser una «invención» es una condición previa para los requisitos: aplicación industrial, novedad y actividad inventiva.

De acuerdo con la tradición en Europa, una «invención» en el sentido del Art. 52(1) CPE debe tener un carácter técnico.

De acuerdo con la jurisprudencia de los Tribunales de Recursos de la OEP el carácter técnico

- a) puede estar en el problema subyacente, o
- b) en los medios (elementos técnicos) formando la solución del problema subyacente, o
- c) en los efectos alcanzados para solucionar el problema, o
- d) puede estar presente si se requieren consideraciones técnicas (o lenguaje técnico) con el fin de

realizar un programa de ordenador.

En efecto, para la mayoría de invenciones implementadas por ordenador, es posible redactar una solicitud de patente y un juego de reivindicaciones que cumplan con este criterio de carácter técnico. Por tanto, en particular para excluir las invenciones de métodos comerciales, la OEP ha desarrollado, desde aproximadamente el año 2000, la teoría de la «contribución técnica» y ha utilizado la condición de actividad inventiva para rechazar invenciones no técnicas. La OEP utilizó la siguiente aproximación problema-y solución para determinar si existía actividad inventiva:

- 1) identificación del sector técnico,
- 2) identificación del estado de la técnica más próximo,
- 3) identificación del problema técnico,
- 4) determinación de las características técnicas de una reivindicación que contribuyen a la solución (la «contribución técnica»).

Una vez se ha verificado la presencia de una contribución técnica, entonces la actividad inventiva se evalúa considerando la reivindicación de su conjunto. Si la contribución proporcionada por la invención tal y como se reivindica es no técnica, por ejemplo está en el sector de la economía, no se toma en consideración en la valoración de la actividad inventiva (PBS Pension Benefits) y la invención no es patentable.

«Estado de la técnica» de acuerdo con el Art. 54 CPE significa «estado de la tecnología» y no se extiende al estado de la técnica en el comercio y en los métodos comerciales. La automatización de un método comercial utilizando hardware y métodos de programación convencionales deben considerarse evidentes para un experto en la materia (Order management/RICOH).

Si una invención relacionada con un programa de ordenador tiene un carácter técnico y proporciona una contribución técnica, es patentable en la forma de un aparato, un método o un programa o un medio de almacenamiento que almacene un programa, que puede leerse por ordenador.

La Comisión y el Consejo de la Unión Europea intentaron armonizar la protección concerniente a las invenciones implementadas por ordenador en la Unión Europea. Su propuesta de Directiva fue rechazada. La práctica nacional en los estados miembros de la Unión Europea continúa. La jurisprudencia alemana y francesa siguen generalmente la jurisprudencia de la OEP. La última jurisprudencia en Gran Bretaña (decisiones Oracle), aunque expresadas en forma ligeramente diferente también está en línea con la jurisprudencia de la OEP.

Resumen

La Oficina Europea de Patentes concede protección por patente a las invenciones relacionadas con programas de ordenador, incluyendo métodos comerciales relacionados con programas de ordenador en la medida en la que tienen carácter técnico y proporcionan una contribución técnica.

3. POSICIÓN DE AIPII Y COMPARACIÓN

3.1. POSICIÓN DE AIPII

La AIPII ha adoptado dos resoluciones en esta materia. La Resolución Q133 fue sobre el tema «Patentabilidad de los programas de ordenador». La Resolución Q158 cubrió el tema «Patentabilidad de los métodos comerciales».

La posición de AIPII en ambas resoluciones se encuentra entre la posición norteamericana de patentar cualquier desarrollo nuevo, no obvio y útil y la posición europea de restringir la protección por patente a los sectores técnicos. Las resoluciones de AIPII se votaron casi por unanimidad después de una intensa discusión. No existió discrepancia entre las tres familias de AIPII (miembros de la industria, agentes y abogados). Es decir, la posición de AIPII claramente muestra la posición internacional de los expertos de propiedad industrial y utilizadores del sistema de patentes en relación con la patentabilidad de las invenciones relacionadas con programas de ordenador.

En la Resolución Q133, la AIPII resolvió que las patentes deberían concederse sin discriminación en todos los sectores de la tecnología, incluyendo el de los programas de ordenador, tales como los programas, y que todos los programas de ordenador que reúnan los requisitos de patentabilidad deberían considerarse patentables de la misma manera y tratados de forma igual, sin distinción entre los diferentes tipos de programas de ordenador y aplicando las mismas reglas que en otros sectores de la tecnología. Además, de acuerdo con la Resolución Q133, los programas de ordenador deberían ser inherentemente patentables en cualquier medio en el que se comercialicen.

En la Resolución Q158, la AIPII resolvió que «los métodos comerciales» deberían ser aptos para la protección por patente siempre y cuando la invención tal y como se define en las reivindicaciones tenga un contenido técnico. De acuerdo con la Resolución Q158, si tal invención en su conjunto tiene un contenido técnico, ello debería ser suficiente para ser patentable a pesar de que la novedad y la actividad inventiva (no obiedad) no recaiga en el contenido técnico. Además, la protección de tales invenciones por patente debería ser valorada o basada en los mismos criterios que otras invenciones, y no deberían aplicarse criterios nuevos o especiales. La mera transformación de un método conocido en forma de programa de ordenador no da lugar a un presunción de que tal invención tiene actividad inventiva.

Por tanto, la formulación de la AIPII requiere contenido técnico como un requisito de patentabilidad, y reconoce como suficiente el contenido técnico de una invención implementada en un sistema técnico (que siempre es el caso de los programas de ordenador), a pesar de que la contribución inventiva no tenga un carácter técnico. De acuerdo con la AIPII, los métodos comerciales nuevos e inventivos y otras innovaciones no técnicas deberían ser patentables si se implementan en un ordenador.

3.2. COMPARACIÓN DE LOS DIVERSOS SISTEMAS

Las diferentes posiciones de las oficinas de patentes más importantes y de la AIPII pueden resumirse por los ejemplos de la siguiente tabla:

Ejemplo de invención		Patentabilidad			
		US	JP	UE	AIPPI
1.	Programa de ordenador implementando un nuevo ciclo de inyección de tiempo variable para el combustible de un motor de automóvil para la optimización del consumo y de la emisión.	si	si	si	si
2 a)	Método para calcular el beneficio diario neto de un fondo de inversión basado en datos de impuestos actualizados e información actualizada del mercado de valores.	si	no	no	no
2 b)	Método según 2.a) implementado en un sistema de ordenador o un programa de ordenador.	si	si*	no**	si

* si las reivindicaciones se redactan para definir una interacción específica entre el software y un recurso de hardware
 ** a menos que la implementación haga una contribución técnica

Como muestra la tabla, la jurisdicción más receptiva para la patentabilidad de los programas de ordenador es la de Estados Unidos y la más restrictiva la de Europa. La posición de Japón, así como la de los expertos internacionales de la AIPPI, está entre medio.

4. BENEFICIOS ECONÓMICOS DE LA PROTECCIÓN POR PATENTE PARA LAS INVENCIONES RELACIONADAS CON PROGRAMAS DE ORDENADOR Y MÉTODOS COMERCIALES

4.1. UTILIDAD DE LAS PATENTES EN GENERAL

La cuestión de si las patentes en general son útiles para la sociedad ha sido debatida durante décadas. Un breve resumen de dicho debate es necesario para tratar la cuestión relativa a las patentes de programas de ordenador y de métodos comerciales. Pero en general la contestación es - Si, es económicamente útil para la sociedad el prever un incentivo a los inventores.

En la mayoría de países europeos, se introdujeron modernos sistemas de patentes en el curso de la industrialización hacia finales del siglo XIX. En aquella época, los beneficios potenciales y los inconvenientes de tal sistema para el interés de la sociedad fueron discutidos en profundidad. Puede observarse que aunque países técnicamente avanzados decidieron adoptar un sistema de patentes, países con un bajo nivel de desarrollo industrial decidieron no conceder protección por patentes. La industria en estos últimos países se aprovechó inicialmente de la posibilidad de utilizar las invenciones de otros. Sin embargo, después de que la industria doméstica en estos países empezara a alcanzar y desarrollar sus propias innovaciones se introdujeron sistemas de patentes para promover y proteger estas innovaciones.

Se han efectuado dos revisiones en profundidad recientemente del sistema de patentes en los Estados Unidos por la Comisión Federal de Comercio de los Estados Unidos (FTC). «To Promote

Innovation. The Proper Balance of Competition and Patent Law and Policy»¹, 2003 y The Board on Science, Technology and Economy Policy, National Academies of Science (NAS) (La Junta de Ciencia, Tecnología y Política Económica de la Academia Nacional de Ciencia). «A Patent System for the 21st Century»², 2004³. Ambas revisiones han concluido que aunque hay numerosas reformas prácticas necesarias que hacer, el sistema de patentes norteamericano para que funcione de forma más efectiva, en general las patentes tienen un papel importante que jugar en la economía moderna. El Informe NAS indica explícitamente que «Índices altos continuados de innovación sugieren que el sistema de patentes está funcionando bien y que no requiere cambios fundamentales»⁴.

En resumen, la justificación económica estándar para las patentes en general, es que una patente proporciona al inventor una mejor oportunidad de recuperar su inversión en desarrollar la invención y así tiene un incentivo para innovar. Este valor social justifica un derecho exclusivo temporal para explotar su nueva invención.

4.2. UTILIDAD DE LAS PATENTES DE INVENCIÓNES IMPLEMENTADAS POR ORDENADOR

Hasta la segunda mitad de los años 80 la cuestión de las implicaciones económicas de las invenciones relacionadas con programas de ordenador no fue seriamente discutida puesto que los programas de ordenador y las transacciones comerciales automatizadas no constituían una parte significativa de la economía global, y puesto que no existía un perjuicio económico perceptible derivado de tal protección de la Propiedad Industrial. Ello ha cambiado enormemente con la introducción de los ordenadores portátiles, comunicaciones y teléfonos móviles, con la digitalización de la mayoría de los sistemas electrónicos (grabaciones de audio y vídeo, radio, televisión, control de máquinas y motores, sistemas bancarios, correo y ventías en línea, etc.). Con la llegada de las redes de ordenadores y de Internet, el sector terciario está creciendo rápidamente en todos los países industrializados, mientras que el sector secundario (producción industrial) se ha movido hacia países menos desarrollados.

En la era anterior a los ordenadores portátiles, los métodos comerciales eran usualmente tratados como secretos industriales y se transmitían de hombres de negocios experimentados a sus sucesores mediante un largo proceso de entrenamiento. Ahora, los métodos comerciales han pasado a ser productos de programas de ordenador que se pueden comprar tales como programas de contabilidad, programas de administración de Propiedad Industrial, sistemas de ventas en línea, sistemas de rebajas, programas de investigación de la clientela en línea, servicios financieros, valores, etc.

El Informe del FTC antes mencionado señala que, en el sector del software, existen problemas, principalmente aquellos relacionados con el otorgamiento de patentes de baja calidad en los sectores de los programas de ordenador e Internet. Sin embargo, ni el FTC ni NAS proponen abolir o recortar severamente la patentabilidad de los programas de ordenador. Al mismo tiempo el Informe NAS revisó la

¹ «Prevention of the Innovation: El equilibrio adecuado de las políticas de competencia y Ley de Patentes»
² «Un sistema de patentes para el Siglo XXI»

historia de la expansión de la ley de patentes norteamericana a nuevas y diversas fronteras tecnológicas (incluyendo los programas de ordenador) e idéntico esto como una de sus principales fortalezas.

Puede observarse que un gran número de patentes de programas de ordenador concedidas por la Oficina Norteamericana de Patentes, la OEP y la Oficina Japonesa de Patentes en los últimos 10 años no ha disminuido la velocidad de innovación en el sector del software. De forma similar, los estudios recientes sobre la innovación en el sector de los servicios financieros muestra que tal innovación no ha disminuido desde que la decisión del caso State Street Bank en los Estados Unidos afirmó la patentabilidad de tales innovaciones si cumplían los requisitos de patentabilidad.

Este incentivo a la innovación dirige las fronteras de toda la ciencia, incluida aquella que depende de los ordenadores. En efecto, un estudio ("The Economic Impact of Patentability of Computer Programs")¹ efectuado por el Instituto de Propiedad Industrial e Intelectual de Londres por encargo de la Comisión Europea en Marzo de 2000 llegó a la conclusión de que "la patentabilidad de las invenciones relacionadas con programas de ordenador ha ayudado al crecimiento de las industrias relacionadas con programas de ordenador en los Estados Unidos, en particular ha ayudado al crecimiento de las PYMES y de los desarrolladores de programas independientes hasta convertirse en compañías de cierto tamaño y en grandes compañías...".

Las industrias de servicios financieros parecen ser altamente innovadoras. Solamente en el área del mercado de valores se ha estimado que en el período 1980-2001 la industria del mercado de valores ha generado entre 1.200 y 1.800 nuevos tipos de títulos. La innovación en el mercado de valores se produce para llenar el vacío de los instrumentos disponibles. Se diseñan constantemente nuevos títulos para esquivar riesgos no evitables de otro modo y para facilitar ingresos procedentes de situaciones que el actual mercado de valores no cubre (lo que los economistas financieros denominan "completar el mercado"). Estudios recientes demuestran que tales innovaciones requieren sustanciales cantidades de mano de obra y capital para desarrollarse. Por ejemplo, se estima que el desarrollo de un nuevo producto financiero requiere una inversión de entre 50.000 y 5 millones de dólares. Esta inversión incluye (a) pagos por asesoramiento legal, contable, regulador y fiscal; (b) tiempo destinado a educar a los emisores, inversores y comercializadores; (c) inversiones en sistemas informáticos sobre evaluación y comercialización; y (d) compromisos de capital y personal para soportar la creación de mercado. Además, los bancos de inversión que innovan pagan habitualmente 1 millón de dólares anuales a personal perteneciente a grupos de desarrollo de productos con dos a seis bancos (Turano, Peler, 1989. Innovación financiera y ventajillas del vanguardista. Revista de Economía Financiera 25 (Diciembre): 213-40). Turano concluye que los bancos de inversión han intentado en el pasado recuperar estas inversiones a través de la reducción de costes en el mercado a través de productos financieros innovadores.

En consecuencia, las importantes inversiones tanto en mano de obra como en capital en estos nuevos instrumentos y sistemas relacionados con las finanzas demandan derechos de propiedad que proporcionen alguna medida que asegure que estas inversiones se recuperaran, para que tales innovaciones en este sector financiero continúen.

¹ El impacto económico de la patentabilidad de los programas de ordenador.

4.3. VENTAJAS DE LAS PATENTES PARA LAS PYMES

Se argumenta con frecuencia que las patentes son ventajosas para las empresas grandes y que amenazan a las empresas pequeñas y medianas (PYMES) o a los inventores individuales. Sin embargo, debe considerarse que la mayoría de las mayores empresas de programas de ordenador ganan un cuasi monopolio en áreas específicas de la industria de los programas de ordenador en un entorno casi libre de patentes durante los últimos 15 a 25 años.

Las grandes empresas son capaces de mantener su dominio exclusivo a base de su poder de mercado. Pueden (y a veces lo hacen) adquirir nuevos desarrollos de programas de ordenador de las PYMES e integrarlos en sus productos estándar existentes. Un usuario de dichos productos estándar probablemente de este modo utiliza este elemento integrado en vez de comprar adicionalmente el producto de programa de ordenador de la PYME innovadora. Esta práctica deja fuera del mercado a las PYMES y únicamente puede evitarse mediante la protección por patentes de dichos nuevos desarrollos.

Los casos recientes en los tribunales norteamericanos muestran que las patentes se utilizan principalmente por las PYMES para defender sus intereses contra las grandes compañías. Las patentes garantizan un retorno adecuado a la inversión de los inventores. Por ejemplo, en los Estados Unidos, muchas universidades ganan al año cerca de un billón de dólares en licencias de patentes lo que equivale a los ingresos por licencias anuales de IBM. Una patente, además, ayuda a las compañías jóvenes a encontrar inversores, muestra que la compañía es innovadora lo que ayuda a recibir encargos de grandes compañías u organizaciones y mejorar las perspectivas de ofertas de compra de la compañía.

4.4. SOFTWARE DE CÓDIGO ABIERTO Y SOFTWARE COMERCIAL

Cuando aparecieron oportunidades de desarrollo técnico así como de desarrollo de programas de ordenador comerciales a finales de los años 70 y principios de los 80, las universidades empezaron a estudiar y a enseñar principios básicos sobre arquitectura de ordenadores y sobre diseño de programas de ordenador, y a entrenar a programadores de ordenadores en estas materias. Ello condujo al desarrollo del movimiento del software de código abierto/libre, que empezó a proporcionar versiones libres o al menos más baratas de los programas de ordenador comerciales mas populares existentes para competir con los desarrolladores de programas de ordenador comerciales (Erwin J. Basinski, "Una breve historia del software": BNA International Inc., World E-Commerce and IP report, Abril 2005, Vol. 5, Nr. 4, p. 24). Por ejemplo, estos desarrollos dieron nacimiento a mucho software libre - BSD (de Bill Joy en UC Berkeley - una versión primera de UNIX a principio de 1980), GNU programas de ordenador (de Richard Stallman y otro en MIT en 1983), LINUX OS (de Linus Torvalds en Europa en 1991) y el servidor Apache y el buscador MOSAIC (de NCSA (Centro Nacional de Aplicaciones de Supercomputación, Universidad de Illinois) a mediados de los 90).

La comunidad del software de código abierto originalmente comprendía a estos académicos, estudiantes y programadores independientes, desarrollo software principalmente sin inversión comercial o protección declarada de Propiedad Industrial e Intelectual. Estos grupos no tenían necesidad de patentes. La naturaleza cooperativa del desarrollo no tenía necesidad de licencias para obtener acceso a la tecnología de otros. De forma similar, debido a que estos desarrolladores del software de código abierto no dependían generalmente de las inversiones exteriores en un grado significativo, no tenían costes de desarrollo significativos que recuperar vendiendo sus productos. Podían facilitar los productos a quien lo

quisiera a bajo o ningún coste.

Estos primeros desarrolladores del software de código abierto se concentraron en programas de ordenador de infraestructura tales como sistemas operativos (Linux y GNU/Unix), servidores (Apache), sistemas de bases de datos (MySQL), buscadores (Mozilla). Programas comerciales similares de fabricantes de hardware informático y desarrolladores de software comercial fueron vendidos por cientos de miles de dólares. Cuando Linux OS, el servidor Apache y otro software informático libre empezó a ser más promovido por los académicos, desarrolladores pequeños y pequeños usuarios, su disponibilidad libre pasó a ser más atractiva a los gestores de negocios de tecnología de la información, todos los cuales afrontaron la presión presupuestaria anual usual para reducir el coste total informático (por ejemplo, los costes de hardware informático más los costes de software, más los costes de mantenimiento). Para estos usuarios existía y todavía existe el temor sobre el software libre en relación con quien podría o lo mantendrá, quien será responsable si no funciona, quien responderá de los daños. Últimamente ha surgido un negocio (Red Hat 1994) desarrollado para mantener y "garantizar" una versión libre de OS/Linux. El servidor Apache fue adquirido por la Fundación Apache en 1999 para soportar el mismo de forma más robusta. Los fabricantes de hardware, liderados por IBM, vieron una oportunidad de pregonar la "reducción del coste total" de sus sistemas de hardware informático adoptando su propia versión de Linux OS y Apache y ofreciéndolo libre pero cobrando por el mantenimiento, copias de seguridad y garantía. IBM & Red Hat anunciaron una alianza con Linux en 1999, y en 2002 anunciaron una alianza plurianual global para proporcionar soluciones Linux empresariales. Debe hacerse notar que estas "soluciones Linux empresariales" podían ser aplicaciones realizadas con software libre combinadas con aplicaciones patentadas por IBM.

Hewlett Packard (HP), Sun y otros principales vendedores de hardware han seguido a IBM adoptando y proporcionando versiones libres de Linux OS y otro software libre, que trabajan con su hardware, con el fin de mantenerse competitivos en costes con IBM y entre sí. Sun ha facilitado incluso una versión de software de código abierto de sus sistemas operativos Solaris (UNIX) con el fin de mantener el coste total informático tan bajo como sea posible. Ello no ha evitado que IBM, HP o Sun hayan continuado con sus programas de patentes de forma vigorosa.

Hoy en día poca gente sabe que muchos, sino la mayoría de los principales contribuidores al éxito comercial de los proyectos de software de código abierto tales como Linux OS, son programadores pagados que están trabajando directa o indirectamente para una entidad comercial (Red Hat, IBM, HP, Novell, Sun, Intel). Por tanto el tema software de código abierto y patentes debe contemplarse en el contexto del software de código abierto entrecruzado en la corriente comercial y financiado de una forma básicamente coherente con la forma como se financia el desarrollo del software habitualmente. IBM o HP, por ejemplo, no son un proveedor "comercial-puro" de software ni tampoco un proveedor de "software de código abierto-puro". Han fusionado conjuntamente ambos modelos. De forma similar, Red Hat puede decirse que abarca ambos modelos. Por lo tanto resulta que hoy en día los programas informáticos de código abierto más significativos utilizados comercialmente están realmente escritos por programadores pagados por las mismas empresas que mantienen algunos de los portafolios de patentes más grandes en el mundo. Sin embargo persiste también una amplia comunidad de académicos, estudiantes y programadores independientes de software libre que están desarrollando versiones libres de programas de aplicaciones que intentan competir con la comunidad del software comercial.

Parece por tanto que la comunidad del software de código abierto/libre está aprendiendo a vivir en un mundo de patentes de programas de ordenador. Los fabricantes de hardware informático más

importantes, que son también los propietarios de los mayores portafolios de patentes de software, han indicado que no tienen la intención de perseguir a los desarrolladores de software de código abierto si utilizan programas sobre los que poseen patentes, ni tampoco están interesados en perseguir a sus clientes. Por el contrario, la sociedad y los gobiernos tendrán multamente que ponerse de acuerdo sobre normas (dirigidas por el mercado o por la sociedad del bienestar u otras) en las cuales los beneficios de los intereses comunes de los usuarios, los desarrolladores de programas comerciales y los desarrolladores de software de código abierto resulten optimizados. Y podría ser que sean necesarios diferentes tipos de normas para los diferentes contextos de las complejas relaciones comerciales descritas anteriormente. Lo que parece claro es que las fuerzas del mercado comercial continuarán a promocionar la innovación y la competencia en el desarrollo de software y productos relacionados con el software. La historia parece demostrar que estos fenómenos mejoran por la protección por patente, y todos los grupos han crecido y prosperado en un entorno de patentes.

CONCLUSIÓN

Las posiciones anteriores de AIPPI sobre las patentes de programas de ordenador (Q133) y sobre patentes de métodos comerciales (Q158) son consistentes con este análisis y deberían reafirmarse.

RESUMEN

Las normas actuales sobre invenciones implementadas por ordenador en los sistemas de patentes más importantes son las siguientes:

- Según la práctica norteamericana, cualquier invención útil hecha por el hombre es patentable, incluyendo cualquier programa de ordenador y método comercial nuevo y no evidente.
- En Japón, las invenciones de programas de ordenador y los métodos comerciales implementados por programas de ordenador son patentables, si se define en las reivindicaciones una interacción específica con un recurso de hardware.
- La Oficina Europea de Patentes concede protección por patentes a invenciones relacionadas con programas de ordenador, incluyendo programas de ordenador relacionados con métodos comerciales, siempre y cuando tengan carácter técnico y realicen una contribución técnica.

Las anteriores Resoluciones de AIPI Q133 y especialmente Q158 requieren un contenido técnico como un prerrequisito de patentabilidad. La AIPI reconoce como suficiente, el contenido técnico de una invención implementada en un sistema técnico (que es siempre el caso de un programa de ordenador), aunque la contribución del inventor no tenga un carácter técnico. De acuerdo con la AIPI, los métodos comerciales nuevos e inventivos y otras innovaciones no técnicas deben ser patentables si se implementan por un ordenador.

La Unión Europea (UE) intentó una armonización de las normas relativas a invenciones relacionadas con programas de ordenador a través de una directiva europea. La directiva europea fue rechazada en julio de 2005. Los puntos más importantes de discusión durante el proceso legislativo en la Unión Europea estuvieron relacionados con la existencia de beneficios económicos de la protección por patentes para las invenciones relacionadas con programas de ordenador y sobre si la patentabilidad constituiría un impedimento a la creatividad y al progreso.

Dos completas revisiones recientes del sistema de patentes norteamericano han sido efectuadas por la Comisión de Comercio Federal de los Estados Unidos (FTC), y por las academias de Ciencias Nacionales (NAS). Ambas han concluido que aunque hay numerosas reformas prácticas necesarias para hacer que el sistema de patentes norteamericano funcione de forma más efectiva, en general las patentes tienen un papel importante que jugar en la economía moderna. El Informe NAS señala explícitamente que: «Los altos índices continuados de innovación sugieren que el sistema de patentes está funcionando bien y no requiere cambios fundamentales». Asimismo, las extensas consultas efectuadas por la Unión Europea en la preparación de la propuesta de la Directiva parecen ir en la misma dirección.

Sin embargo, coexiste con la industria de programas de ordenador comerciales que se aprovecha del sistema de patentes una gran y creciente comunidad de software de código abierto, que desarrolla programas de ordenador principalmente sin inversión comercial. Aquellos que trabajan en esta comunidad podrían tener poca o ninguna necesidad de patentes. Sin embargo, el marco económico del software de código abierto está cambiando. Fabricantes de hardware tales como IBM han adoptado software de código abierto y facilitan versiones libres de Linux OS y otro software libre, que trabaja con su hardware con el fin de mantenerse competitivos en costes.

Expediente No 03-46.838

Solicitud de registro para la patente titulada COMPUTACIÓN DISTRIBUIDORA
BASADA EN IDENTIDAD PARA RECURSOS DE ARTEFACTOS

Solicitante: Microsoft Corporation.

Tomo 1

TÍTULO DE PATENTE NO. 240176

Titular(es):	MICROSOFT CORPORATION
Domicilio(s):	One Microsoft Way, Redmond, Washington, 98052, E.U.A.
Denominación:	COMPUTACION DISTRIBUIDA BASADA EN IDENTIDAD PARA RECURSOS DE DISPOSITIVO
Inventor(es):	DAVID ALLEN GEORGE M. MOORE
Número:	PA/2003/004409
País:	US
SOLICITUD	PRIORIDAD
Fecha de presentación:	19 de mayo de 2003
Horas:	15:18
Número:	20 048 JUNIO 19 2002

ESTA PATENTE CONCEDE A SU TITULAR EL DERECHO EXCLUSIVO DE EXPLOTACIÓN DEL INVENTO RECLAMADO EN EL CAPÍTULO DE INVENTARIO. X TIENE UNA VIGENCIA IMPROPRIOGABLE DE VEINTE AÑOS CONTADOS A PARTIR DE LA FECHA DE PRESENTACIÓN DE LA SOLICITUD.

Fecha de expedición: 11 de septiembre de 2006

EL DIRECTOR DIVISIONAL DE PATENTES

QUIM. FABIAN R. SALAZAR GARCÍA



PROTESTO A USTED MI MAS ATENTA CONSIDERACION
México, D.F., Septiembre 11, 2006
POR MICROSOFT CORPORATION

APODERADO
ING. JAVIER SAUCEDO C.

ANEXOS. Boleta de pago por el título, próroga y las anualidades correspondientes
DOMICILIO PARA OIR NOTIFICACIONES. Pedro Luis Ogazón 17 Col. San Angel
01000 México, D.F.
CRM/mmc03P0591

SEGUNDO. - Se siiva ordenar se continúe con la tramitación de esta solicitud y se conceda a mi mandante la protección solicitada en base a la nueva documentación que se acompaña.

PROTESTO A USTED MI MAS ATENTA CONSIDERACION

México, D.F., a 2 de junio de 2006

POR MICROSOFT CORPORATION

APODERADO

ING. JAVIER SAUCEDO C.

Anexos: Nuevo capítulo reivindicatorio y resumen, por triplicado.

DOMICILIO PARA OIR NOTIFICACIONES: Pedro Luis Ogazón No. 17
Col. San Ángel
01000, México, D.F.

CRM03P0591

REIVINDICACIONES

1. En un entorno de computación, un método que comprende:
en un primer dispositivo de computación, descubrir un
segundo dispositivo de computación con base en recursos que son
accesados en base a un servicio de seguridad .NET;
en el primer dispositivo de computación, obtener datos
relacionados con el segundo dispositivo de computación, incluyendo los
datos la información de los recursos del segundo dispositivo de
computación que se formatea de conformidad con un esquema que
entiende el primer dispositivo de computación; e
interpretando el primer dispositivo de computación los datos
para acceder un recurso del segundo dispositivo de computación.
2. El método de la reivindicación 1 que comprende
además, comunicar una solicitud para usar el recurso desde el primer
dispositivo al segundo dispositivo.
3. El método de la reivindicación 1 que comprende
además, comunicar la información de identidad desde el primer dispositivo
al segundo dispositivo.
4. El método de la reivindicación 3, en donde la
información de identidad corresponde a credenciales.
5. El método de la reivindicación 3, en donde las
credenciales se proporcionan mediante un servicio de autenticación.
6. Un medio legible por computadora que tiene
instrucciones ejecutables por computadora para realizar un método, el

método que comprende:

en un primer dispositivo de computación:
localizar la información acerca de un segundo dispositivo de
computación que está disponible para usarse, la información localizada
que se basa en una identidad basada en servicios de seguridad .NET
asociados con el primer dispositivo de computación;
seleccionar un recurso del segundo dispositivo a partir de la
información; y

accesar el recurso del segundo dispositivo de computación.

7. En un entorno de computación, un sistema que
comprende:

un primer dispositivo de computación asociado con cuando
menos una identidad basada en servicios de seguridad .NET, incluyendo el
primer dispositivo de computación un primer recurso y configurado para
proporcionar acceso al primer recurso mediante cuando menos otro
dispositivo que se asocia con la identidad, proporcionando el primer
dispositivo de computación información acerca del primer recurso de
conformidad con un esquema de recurso que se asocia con la cuando
menos una identidad;

un segundo dispositivo de computación, el segundo dispositivo
de computación que se configura para solicitar acceso al primer recurso; y
el segundo dispositivo de computación realizando una tarea
de computación que incluye usar el primer recurso en el primer dispositivo
de computación, para realizar cuando menos parte de la tarea.

8. El sistema de la reivindicación 7, en donde el primer

recurso comprende almacenamiento.

9. El sistema de la reivindicación 7, en donde el primer
recurso comprende el ancho de banda de una conexión a partir del primer
dispositivo a cuando menos otro dispositivo de computación.

10. El sistema de la reivindicación 7, en donde el primer
recurso comprende ciclos de procesamiento de un procesador.

11. El sistema de la reivindicación 7, en donde el primer
recurso comprende un método de entrada.

12. El sistema de la reivindicación 7, en donde el primer
recurso comprende un método de salida.

13. El sistema de la reivindicación 7, que comprende
además un tercer dispositivo de computación que se conecta al primer
dispositivo de computación, en donde el segundo dispositivo accesa el
primer recurso para realizar una operación con el tercer dispositivo de
computación.

14. El sistema de la reivindicación 7, en donde el tercer
dispositivo de computación comprende una impresora, y en donde el
primer recurso corresponde a un controlador de impresora en el primer
dispositivo de computación.

15. El sistema de la reivindicación 7, en donde el tercer
dispositivo de computación comprende un servidor, y en donde el primer
dispositivo de computación actúa como un proxy para el servidor para el
segundo dispositivo de computación.

16. El sistema de la reivindicación 15, en donde el ancho de
banda del segundo dispositivo de computación y el primer dispositivo de

computación es mayor que el ancho de banda del primer dispositivo de computación para el tercer dispositivo de computación.

17 Un medio legible por computadora que tiene almacenada en el mismo una estructura de datos, la estructura de datos que comprende:

un primer campo para identificar un tipo de recurso;

un segundo campo para identificar un dispositivo que incluye un recurso que corresponde al tipo de recurso que se identificó en el primer campo.

10 un conjunto de cuando menos un tercer campo, correspondiendo cada tercer campo en el conjunto al segundo campo para identificar la información acerca del recurso de dispositivo que se identificó mediante el primer campo;

un conjunto y de cuando menos un cuarto campo, correspondiendo cada cuarto campo en el conjunto a una identidad de usuario con base en servicio de seguridad .NET; y

en donde la estructura de datos se usa para proporcionar información de recursos a otro dispositivo que sea diferente del que se identificó en el segundo campo, de manera que el otro dispositivo pueda usar un recurso que se identificó en el conjunto del tercer campo.

18 La estructura de datos de la reivindicación 17, en donde el primer campo incluye la información que corresponde a un tipo de recurso de almacenamiento.

19 La estructura de datos de la reivindicación 18, en donde al conjunto de tercer campo incluye información que corresponde a los

datos de tamaño total del recurso de almacenamiento.

20 La estructura de datos de la reivindicación 18, en donde el conjunto de tercer campo incluye información que corresponde a los datos de espacio libre del recurso de almacenamiento.

21 La estructura de datos de la reivindicación 17, en donde el primer campo incluye información que corresponde a los datos del ancho de banda.

22 La estructura de datos de la reivindicación 17, en donde el primer campo corresponde a la energía de procesamiento.

23 La estructura de datos de la reivindicación 22, en donde el conjunto de tercer campo incluye información que corresponde a los datos de velocidad de un procesador.

24 La estructura de datos de la reivindicación 22, en donde el conjunto de tercer campo incluye información que corresponde a los datos de uso de un procesador.

25 La estructura de datos de la reivindicación 22, en donde el conjunto de tercer campo incluye información que corresponde a los datos de tamaño de una memoria.

26 La estructura de datos de la reivindicación 25, en donde el conjunto de tercer campo incluye información que corresponde a los datos de uso de una memoria.

27 La estructura de datos de la reivindicación 17, en donde el segundo campo incluye información que identifica el dispositivo como una computadora personal.

28 La estructura de datos de la reivindicación 17, en donde

el segundo campo incluye información que identifica el dispositivo como una unidad de disco.

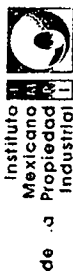
29. La estructura de datos de la reivindicación 17, en donde

el segundo campo incluye información que identifica el dispositivo como
5 un teléfono.

RESUMEN

Un sistema, método y estructuras de datos relacionados para descubrir y describir recursos de computación disponibles en diferentes dispositivos de computación, y para exponer aquellos recursos como

- 5 servicios que son dirigibles mediante aplicaciones de software. Los datos que describen los recursos se configuran de acuerdo a un esquema que se base en identidad. Los recursos de computación pueden incluir, por ejemplo, capacidad de almacenamiento, ancho de banda, energía de procesamiento, métodos y mecanismos de entrada, y métodos de presentación. El método y sistema se basan en identidad, mediante lo cual un usuario (con una identidad) tiene acceso a los recursos distribuidos en proporción a esa identidad.
- 10



"2006, Año del Bicentenario del natalicio del Benemérito de las Américas, Don Benito Juárez García"

DIRECCION DIVISIONAL DE PATENTES
SUBDIRECCION DIVISIONAL DE EXAMEN DE FONDOS DE PATENTES
COORDINACION DEPARTAMENTAL DE EXAMEN DE FONDO AREA ELECTRICA

Asunto: Se comunica el resultado del examen de fondo

México, D.F., a 24 de enero de 2006

SERGIO L. OLIVARES RODRIGUEZ

Apoderado de

MICROSOFT CORPORATION.

Pedro Luis Ogazón 17

San Angel

01000. Alvaro Obregón, Distrito Federal

Su sollicitud No. P/A/2003MOM-69 de 8 marzo enmendado - L.O. 7/2003

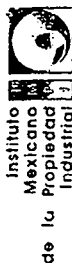
REF:

Como resultado del examen de fondo, realizado con fundamento en los artículos 53 de la Ley de la Propiedad Industrial (LPI) y 42 del Reglamento de la Ley de la Propiedad Industrial (RLPI), se le comunica lo siguiente:

El examen fue realizado en base a los siguientes documentos de la sociedad:

Descripción: No. 1 al 47., como originalmente fueron presentadas.

Figuras: No. 1 al 7 como originalmente fueron presentadas.



er el documento.

Documento 2: US 6 078 955 (KONNO CHISA IO ET AL), 20 de Junho de 2000.

El cual describe la invención reivindicada. En la columna 4 líneas 52 a la 57, en la columna 5 líneas 32 a la 39, en la columna 5 líneas 62 a la 63, en la columna 6 líneas 1 a la 26 y figura 5 del documento 2, y afecta la novedad de las reivindicaciones 17 a la 29.

Por lo anterior, se le requiere que las partes que carecen de novedad, se redacten como estado de la técnica y se caractericen, lo que sea novedoso.

se anexan copias de los documentos citados.

De acuerdo al estado de la técnica citada, se concluye que la materia reclamada en las reivindicaciones 1 a la 29 carecen de novedad, no cumpliendo con lo dispuesto en los artículos 16 y 12 (adición I) de la LPI. Por lo anterior, se le requiere al solicitante de conformidad con el artículo 45 del RLPI, que realice las aclaraciones pertinentes respecto a la novedad de su invención.

Las adiciones o modificaciones realizadas ya sea en la descripción, en los dibujos y/o en las reivindicaciones, no deberán contener materia adicional con mayor alcance que la materia presentada originalmente en la solicitud y/o elementos que den soporte a reivindicaciones adicionales, de tal manera que se cumpla con lo establecido en el artículo 55 BIS LPI.

simismo deberá efectuar el pago que establece la tarifa vigente y exhibir el comprobante correspondiente.

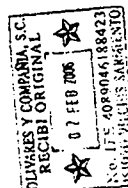
Para que satisfaga estos requisitos, se le concede un plazo de dos meses contados a partir del día siguiente a la fecha en que se le notifique el presente oficio, apercibido que de no cumplir este requerimiento en el plazo señalado se considerará abandonada su solicitud de patente de conformidad con lo dispuesto por los arts. 55 y 58 de la Ley de la Propiedad Industrial.

[illegible]

4 TENTAMENTE

COORDINADOR DEPARTAMENTAL

EDRO DAVID FRAGSITOE7



250027.djvu

[illegible]

2

INSTITUTO MEXICANO DE
LA PROPIEDAD INDUSTRIAL

Dirección Divisional de Patentes

C.P. 060700/84485
C.P. 060700/84485
C.P. 060700/84485
C.P. 060700/84485



Solicitud No. PA/a/2003/004409

de Patente

Sección Prioridades

009

C. DIRECTOR GENERAL DEL INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL

Me refiero al asunto mencionado al rubro, para manifestar a usted que en cumplimiento de lo dispuesto en el Artículo 41 de la Ley de la Propiedad Industrial y 36 de su Reglamento, y para el efecto de que se reconozca a mi mandante el derecho de prioridad solicitado, acompaño al presente escrito una copia de la solicitud Norteamericana No. 10/187.221 debidamente certificada por la Oficina de Patentes correspondiente y traducida al español.

En esta virtud, ruego a usted se sirva tener por exhibidos los documentos que se anexan para todos los efectos legales a que haya lugar, y reconocer a mi mandante el derecho de prioridad solicitado.

Protesto a usted mis respetos.

México, D.F., a 19 de Agosto del 2003.

POR MICROSOFT CORPORATION

APODERADO
Ing. Javier Saucedo C.

ANEXOS: Copia certificada de la solicitud Norteamericana No. 10/187.221 y su traducción al español.

DOMICILIO PARA OIR NOTIFICACIONES: Pedro Luis Ogazón 17
Col. San Ángel, C.P. 01000
Delegación Álvaro Obregón
México, D.F.

VSH*



2003
Don Miguel Hidalgo y Costilla, Padre de la Patria

de la

DIRECCION DIVISIONAL DE PATENTES
SUBDIRECCION DIVISIONAL DE PROCESAMIENTO ADMINISTRATIVO DE PATENTES
COORDINACION DEPARTAMENTAL DE EXAMEN DE FORMA
Expediente de Patente de Invención Normal PA/a/2003/004409

Asunto: Requisitos de Forma

México, D.F., a 4 de agosto de 2003
No. de Folio: 37655

SERGIO L. OLIVARES RODRIGUEZ

ApoDERado de

MICROSOFT CORPORATION

Pedro Luis Ogazón 17

San Ángel

01000, Álvaro Obregón, Distrito Federal

RECIBI ORIGINAL

Nombre:

Fecha: 17 SEP 2003

Firma:

REF: Se da acuse de recibo a su Solicitud presentada el 19/05/2003.

Deberá cumplir con los siguientes requisitos:

16. Presentar documento de cesión de derechos. (art. 39º de la Ley y art. 5º fracción VIII del Reglamento).

35. Entregar la cantidad por concepto de subanación de omisiones que establece el artículo 29 de la Tarifa por los Servicios que presta el Instituto Mexicano de la Propiedad Industrial.

Para cumplir con lo señalado en el art. 50 de la Ley, se le concede un plazo de dos meses contados a partir del día siguiente de la fecha en que se le notifique el presente oficio. Con fundamento en el art. 58 de la Ley, se le otorga un plazo adicional de dos meses, contados a partir del día siguiente al del vencimiento del primer plazo, para cumplir con dichos requisitos, sin que medie escrito y comprobando el pago de la tarifa que corresponda al mes en que se de cumplimiento, apercibido que su solicitud presentada ante este Instituto se tendrá por abandonada si no cumple en el plazo inicial o en el adicional o no presenta el comprobante de pago de las tarifas correspondientes.

El suscrito firma el presente oficio con fundamento en los artículos 8º fracciones III, VI y VII y 7º bis 2 de la Ley de la Propiedad Industrial, 1º, 3º, 4º, 7º fracciones III, V, IX, XV y XVI del Reglamento de la Ley de la Propiedad Industrial, 1º, 3º, 4º, 7º fracciones III, V, IX, XV, 12 fracción II, 15 y 18 fracciones I, II, III, IV y VI del Estatuto Orgánico del Instituto Mexicano de la Propiedad Industrial (D.O.F. 27/12/1999), 1º, 2º y 5º Indefinidos I) y penúltimo párrafo del acuerdo que delimita las facultades en los Directores Generales, Adjuntos, Directores Divisionales, Titulares de las Oficinas Regionales, Subdirectores Divisionales, Coordinadores Departamentales y otros subalternos del Instituto Mexicano de la Propiedad Industrial (D.O.F. 04/02/2000).

A T E N T A M E N T E
EL COORDINADOR DEPARTAMENTAL

C. RAFAEL CASTRO MEJIA

RESEMPLEADO

Nota: Al suscrito este Oficio deberá hacer referencia al No. de Folio y al No. de Expediente correspondiente.



PA/a/2003/004409

368

INSTITUTO MEXICANO DE
LA PROPIEDAD INDUSTRIAL
Dirección Divisinal de Patentes

Expediente: PA/27882/2003
Fecha: 17/02/2003 Hora: 17:18
Código de barras: 001-18-PA/27882/2003



Código de Promoción No. 001

Expediente de Patente de Invención Normal
No. PA/2003/004409
Dirección Divisinal de Patentes
Subdirección Divisinal de Procesamiento
Administrativo de Patentes
Coordinación Departamental de Examen de Forma
REEMFND/2003

C. DIRECTOR GENERAL DEL INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL

No refiero a la solicitud de patente que a nombre de MICROSOFT CORPORATION se tramita en el expediente identificado al rubro, para manifestar a usted lo siguiente:

Que de conformidad con los requisitos contenidos en su oficio Folio No. 37655 del 4 de agosto de 2003, recibido el 17 de septiembre de 2003, se presenta el documento de Cesión de Derechos con el fin de acreditar el carácter del causahabiente de la solicitud arriba identificada.

En esa virtud, a usted C. Director, atentamente pido:

- I.- Tener por contestado en tiempo y forma su aliento oficio; y
- II.- Se continúe el trámite de la presente solicitud en base a la documentación que se acompaña.

PROTESTO A USTED MI MÁS ATENTA CONSIDERACIÓN
México, D. F., 14 de noviembre de 2003.
Por: MICROSOFT CORPORATION

Apoderado
Ing. Javier Saucedo C.

Auxes. Documento de Cesión de Derechos
Comprobante del pago de derechos

DOMICILIO PARA OIR NOTIFICACIONES: Pedro Luis Ogazón No. 17, Col. San Angel
01000 México, D. F.

Dmm*/03AP05/01

INSTITUTO MEXICANO DE
LA PROPIEDAD INDUSTRIAL
Dirección General de Patentes

Expediente: PA/27882/2003
Fecha: 17/02/2003 Hora: 15:47
Código de barras: 001-18-PA/27882/2003



019
Solicitud de Patente No. PA/2003/004409
DEPARTAMENTO DE EXAMEN DE FONDO
AREA ELECTRICA
TITULACION Y CONSERVACION DE PATENTES
ANUALIDADES 2006 A 2010

INSTITUTO MEXICANO DE LA PROPIEDAD INDUSTRIAL

ING. JAVIER SAUCEDO C., en representación de MICROSOFT CORPORATION, se tramita en el expediente mencionado al rubro, y en particular a su aliento oficio No. 45919 del 22 de junio de 2006, recibido el 3 de julio de 2006, cuyo término para contestación venció el día 3 de septiembre de 2006, estando dentro del plazo adicional de un mes que proporciona el Artículo 58 de la Ley en vigor, para manifestar a usted lo siguiente:

Que vengo a enterar los derechos correspondientes a las anualidades 4ª a la 8ª del asunto de referencia, a fin de prorrogar la vigencia del mismo hasta el año 2010.

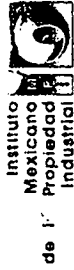
A efecto, proporciono los datos siguientes:

TITULAR: MICROSOFT CORPORATION
FECHA LEGAL: MAYO 19, 2003
FECHA DE EXPEDICION: SEPTIEMBRE 11, 2006
ANUALIDADES: DE LA CUARTA A LA OCTAVA

Supuesto lo anterior,

A USTED C. DIRECTOR GENERAL, atentamente pido se sirva:

- I. Tenerme por presentado en los términos de este escrito; y
- II. Aceptar el pago de las anualidades correspondientes, que prorrogarán la vigencia de este registro hasta el año 2010.



Instituto
Mexicano
de Propiedad
Industrial

Certificado de acuse
de recibo registrado:
PA/2003/033196;
PA/2003/048113
DIRECCION DIVISIONAL DE PATENTES
SUBDIRECCION DIVISIONAL DE PROCESAMIENTO ADMINISTRATIVO DE
PATENTES
COORDINACION DEPARTAMENTAL DE EXAMEN DE FORMA
Expediente de Patente de Invención Normal PA/02/003/004409

Asunto: Resultado del Examen de Forma y Reclamo de Prioridad.

México, D.F., a 16 de agosto de 2004

No. de Folio: 52295

SERGIO L. OLIVARES RODRIGUEZ

Apoderado de
MICROSOFT CORPORATION*

Pedro Luis Oquazón 17

San Angel,

01000, Álvaro Obregón, Distrito Federal

RECIBI ORIGINAL

Nombre:

Fecha:

Firma:

REF: Se da acuse de recibo a su(s) escrito(s) presentado(s), 19/02/2003, 14/11/2003.

Se tiene por satisfecho el examen de forma señalado por la Ley de la Propiedad Industrial y su Reglamento.

Se toma nota de la(s) siguiente(s) prioridad(es) (País, No. de Serie, Fecha de Presentación):

US - 10/187,221 de fecha 28 de junio de 2002,

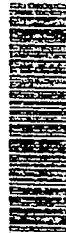
así como su(s) traducción(es) al Español.

El suscrito firma el presente oficio con fundamento en los artículos 6º fracciones III, XI y XII y 7º bis 2 de la Ley de la Propiedad Industrial, 1º, 3º, 4º, 7º fracciones III, V, IX, XV y XVI, 8º fracciones I, II, III, IV y V, 11, 12 fracciones I, II, III, IV y VI del Reglamento del Instituto Mexicano de la Propiedad Industrial (D.O.F. 14/12/1999), 1º, 3º, 5º, 11 fracciones III, V, IX y XVI, 12 fracción II, 15, 16 fracciones I, II, III, IV y VI y 30 del Estatuto Orgánico del Instituto Mexicano de la Propiedad Industrial (D.O.F. 27/12/1999); 1º, 2º y 5º incisos e) y penúltimo párrafo del acuerdo que delega facultades en los Directores Generales Adjuntos, Directores Divisionales, Titulares de las Oficinas Regionales, Subdirectores Divisionales, Coordinadores Departamentales y otros subalternos del Instituto Mexicano de la Propiedad Industrial, (D.O.F. 04/02/2000).

ATENTAMENTE
EL COORDINADOR DEPARTAMENTAL

C. RAFAEL ESTRADA MEJIA

REEMITIDO



Patente 2004

365



US005898870A

United States Patent [19]

[11] Patent Number: 5,898,870

Okuda et al.

[45] Date of Patent: Apr. 27, 1999

[54] LOAD BALANCING FOR A PARALLEL COMPUTER SYSTEM BY EMPLOYING RESOURCE UTILIZATION TARGET VALUES AND STATES

5,630,129 5/1997 Wheat 395,675

Primary Examiner—Lance Leonard Barry
Attorney, Agent, or Firm—Antonelli, Terry, Stout & Kraus, LLP

[75] Inventors: Tooru Okuda, Fujisawa; Yoshiyuki Iwakura, Yokohama; Hirofumi Nagasuka, Kawasaki, all of Japan

[57] ABSTRACT

[73] Assignee: Hitachi, Ltd., Tokyo, Japan

[21] Appl. No.: 08/766,853

[22] Filed: Dec. 13, 1996

[30] Foreign Application Priority Data

Dec. 18, 1995 [JP] Japan 7-328442

[51] Int. Cl.⁶ G06F 15/00

[52] U.S. Cl. 395/674; 395/675

[58] Field of Search 395/674, 675

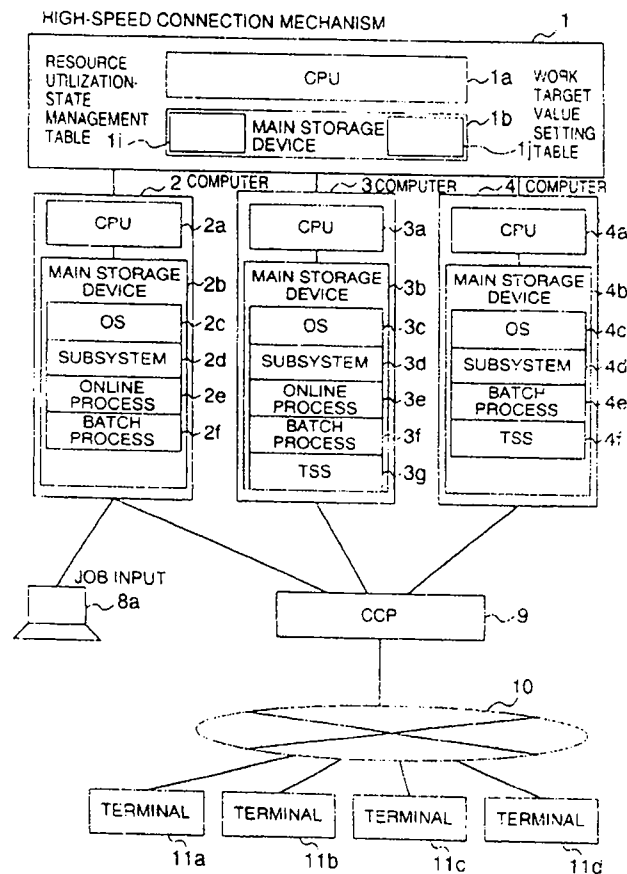
[56] References Cited

U.S. PATENT DOCUMENTS

4,099,235 7/1978 Hoschler et al. 364/200
5,307,496 4/1994 Ichinose et al. 395/650
5,475,844 12/1995 Shiramizu et al. 395/700

A load sharing method for a parallel computer system having a computer group including a plurality of computers and an operation management mechanism which is a computer for managing the operation of the computer group. The method shares a load for executing a plurality of kinds of work processes to the plurality of computers in the computer group, and includes the steps of setting resource utilization target values by work for the plurality of computers in the computer group; collecting resource utilization states by work for the plurality of computers in the computer group to thereby inform the operation management mechanism of the resource utilization states; selecting a computer to execute a newly requested work process from the plurality of computers in the computer group on the basis of the differences between resource utilization target parameter values by work in the plurality of computers in the computer group and current values of a parameter indicating the reporting resource utilization states by work; and executing, in the selected computer, the newly requested work process.

11 Claims, 5 Drawing Sheets



LOAD BALANCING FOR A PARALLEL COMPUTER SYSTEM BY EMPLOYING RESOURCE UTILIZATION TARGET VALUES AND STATES

BACKGROUND OF THE INVENTION

The present invention relates to a load sharing control method for controlling load sharing in a parallel computer system, and particularly to an effective technique to be applied to a load sharing control method for sharing a multiple of kinds of jobs or sessions among a plurality of computers in a parallel computer system with good balance to improve processing speed and guarantee of the turn-around time of jobs.

As main jobs or sessions executed in a conventional computer system, there are jobs or sessions performed by batch process, online process and TSS (timesharing system) process. The jobs, sessions or transactions are hereinafter referred to as "works". In order to operate these works efficiently in one computer system, the order of priority for execution of the works is set by parameters.

On the other hand, the idea of executing a plurality of works in a parallel computer system having a plurality of computers connected in parallel has been proposed to increase the amount of works to be executed. Upon such a background, some methods or apparatuses have been proposed as load sharing control methods or load sharing control apparatuses each used in a parallel computer system for sharing a load among a plurality of computers executing a plurality of works.

JP-A-3-19035 discloses a job execution multiplicity control system in which the job execution multiplicity of a job execution queue as a whole is adjusted to attain increased speed of a batch job while the computer system is in operation.

In brief, the job execution multiplicity control system comprises a number-of-jobs recognition means for recognizing the number of currently running jobs from the job execution queue as a whole, a job execution multiplicity recognition means for making the computer system recognize designated job execution multiplicity, and a job control means for controlling job extraction.

JP-A-5-197682 discloses an online job dynamic load control system in which the lowering of online response can be prevented effectively even in the case where dynamic load in online transaction increases.

In brief, the dynamic load control system comprises a load state recognition means for measuring load on a central computer at predetermined timing to recognize the state of load, and a starting control means for changing/controlling the processing multiplicity of sessions to be executed only when the measured value of load exceeds a predetermined value.

Further, JP-A-3-257654 discloses a load sharing system using session multiplicity in which sessions are assigned to processors in a range of session largest multiplicity to thereby reduce an overhead of a load sharing process per se to improve efficiency of a computer system as a whole.

In brief, when the number of currently operating sessions on a multiplicity information table for storing the session largest multiplicity and the number of currently operating sessions correspondingly to the respective processors does not exceed the largest multiplicity, a session requested to be started is assigned to and started by one of the processors which is determined correspondingly to the session largest multiplicity.

Further, JP-A-6-19861 (U.S. Pat. No. 5,504,894) discloses a transaction processing server path setting mechanism in which response time of online transaction is set by a user by the job class so that the transaction is shared to achieve the response time.

In brief, in a multiprocessor transaction processing system, a transaction is classified into a plurality of classes in accordance with the processing time required therefor. target response time is set by the class and load on each transaction processing server is adjusted so that response performance becomes best in relation between the target response time and actual processing time with respect to all the classes.

As a result of study of the aforementioned prior art references, the present inventors have found the following problems.

That is, in the load sharing control method disclosed in the first-mentioned prior art reference, the load sharing control method is carried out in only one utilization mode in which one kind of works are executed. Accordingly, there is a problem that load sharing adapted to other utilization modes in which a plurality of kinds of works are executed by batch process, online process, TSS process, etc. cannot be carried out.

Further, in the load sharing control method according to the aforementioned prior art, in order to make it possible to share load in accordance with the utilization mode for executing a plurality of kinds of works, load sharing in accordance with an operation mode is carried out in a manner so that the number of initiators (the number of sessions), which indicates multiplicity and which is defined in each sub-system for executing a specific one kind of works is set so as to be different from the number of initiators in another sub-system.

It is however required that a plurality of sub-systems for executing a specific kind of work processes are required to take charge of other kinds of works correspondingly to the characteristic thereof to hereby attain the increase in processing speed.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a load sharing control method and apparatus in which the load for a plurality of kinds of works can be assigned to a plurality of computers constituting a computer group in accordance with the utilization modes.

Another object of the present invention is to provide a load sharing control technique in which the load can be shared among a plurality of computers constituting a computer group correspondingly to the respective characteristics of kinds of works.

A further object of the present invention is to provide a load sharing control technique in which the load can be shared uniformly when the load for a plurality of kinds of works is assigned to a plurality of computers constituting a computer group.

Typical aspects of the present invention disclosed in this specification will be described below in brief.

According to a first aspect of the present invention, in a parallel computer system having a computer group constituted by a plurality of computers, and an operation management mechanism which is a computer for managing the operation of the computer group, a load sharing control method for sharing load for executing a plurality of kinds of work processes to the plurality of computers in the computer

368

group, comprises the steps of: setting resource utilization target values by work for the plurality of computers constituting the computer group; collecting resource utilization states by work for the plurality of computers constituting the computer group to thereby inform the operation management mechanism of the resource utilization states; selecting a computer to execute a newly requested work process from the plurality of computers in the computer group on the basis of the differences between resource utilization target parameter values by work in the plurality of computers constituting the computer group and current values of a parameter indicating the reported resource utilization states by work; and executing, in the selected computer, the work process newly requested to be started.

In the above load sharing control method, first, resource utilization target values by work in respective computers constituting a computer group are set in advance in the operation management mechanism by using parameter definitions, or the like.

Then, the plurality of computers constituting the computer group collect resource utilization states by work in the respective computers and inform the operation management mechanism of the collected individual resource utilization states by work.

The operation management mechanism has a resource utilization-state management table which is prepared in the initialization of the system. Resource utilization states by work informed from the plurality of computers constituting the computer group are arranged on the resource utilization-state management table, so that the respective computers recognize the resource utilization states of the plurality of computers constituting the computer group by referring to the resource utilization-state management table.

In response to a starting request by any computer to newly start a work process in the parallel computer system having the operation management mechanism and the computer group constituted by the plurality of computers, the request issuing computer constituting the computer group compares the resource utilization target value of each work (process) with the resource utilization state thereof on the resource utilization-state management table for each of the group computers. Thus, one of the computers which is to execute a newly requested work process is selected from the plurality of computers constituting the computer group on the basis of the result of the comparison.

Each computer communicates with the selected computer and instructs the selected computer to execute the work process newly requested to be started. Thus, the selected computer executes the newly requested work process.

In the above load sharing control method, in this manner, the operation mode of the parallel computer system as a whole can be defined by setting in advance resource utilization target values by work, so that load sharing can be achieved in accordance with the operation mode which is defined in advance.

Further, in the above load sharing control method, turnaround time can be guaranteed at the time of execution of a work process corresponding to the utilization target value in a manner so that resources corresponding to the preliminarily designated resource utilization target value is made free in advance.

According to a second aspect of the present invention, in the load sharing control method according to the first aspect of the present invention, the parameter indicating the resource utilization states by work of the plurality of computers constituting the computer group informed from the

plurality of computers constituting the computer group to the operation management mechanism contains at least one of CPU utilization ratio information of each of the computers constituting the computer group, CPU utilization ratio information weighted by difference in performance of the plurality of computers constituting the computer group, and main storage free-state information in the plurality of computers constituting the computer group.

In the above load sharing control method, the respective computers constituting the computer group collect the CPU utilization ratio which is the proportion of the processing time per unit time of the central processing units in the respective computers constituting the computer group, the CPU utilization ratio which is weighted by differences in performance of the respective computers constituting the computer group, or the main storage free-state which is the proportion of the number of free pages in the main storage devices of the respective computers constituting the computer group, as the resource utilization states of the plurality of computers constituting the computer group to be informed to the operation management mechanism.

Here, the CPU utilization ratio, which is weighted by differences in capability of the respective computers constituting the computer group, means the relative CPU utilization ratio which is given when the processing performance capability (e.g. expressed by "FLOPS") of a computer lowest in processing (capability) among the plurality of computers constituting the computer group is assumed to be "1".

In the parallel computer system having the operation management mechanism and the computer group constituted by the plurality of computers, the works requested to be started include works having various characteristics, such as works very high in load on the central processing unit, works requiring large main storage areas, and so on.

Therefore, in the above load sharing control method, each of the computers constituting the computer group collects the information concerning the utilization states of a plurality of kinds of resources as the resource utilization states of the respective computers constituting the computer group, so that load sharing is performed correspondingly to the collected resource utilization states.

As described above, according to the above load sharing control method, because the utilization states of a plurality of kinds of resources which are actually used in the plurality of computers constituting the computer group are collected so that a newly requested work process is executed by one of the computers selected in accordance with a specific resource utilization state, the load can be shared among the plurality of computers constituting the computer group correspondingly to the respective characteristics of the plurality of kinds of works.

In the load sharing control method according to the first or second aspect of the present invention, the step of selecting a computer to execute a newly requested work process includes a substep of calculating a difference between a resource utilization target parameter value by work in each of the computers constituting the computer group and the informed resource utilization state parameter by work to thereby select a computer in which the difference is largest.

In the parallel computer system having the operation management mechanism and the computer group constituted by the plurality of computers, when a request to start a work process is newly given and a computer which executes the work process is to be selected, if one of the

computers which is small in difference between the resource utilization target value by work in each of the computers constituting the computer group and the informed resource utilization state by work is selected to make the selected computer execute the work process, the load on the selected computer may be increased so that the completion of the processing may be delayed.

In the above load sharing control method, therefore, when a computer which is to execute a work process newly requested to be started is selected, the resource utilization target values by work in the respective computers constituting the computer group are compared with the reported resource utilization states by work so that one of the computers largest in difference is selected from the result of the comparison so as to execute the work process newly requested to be started. Accordingly, when load for a plurality of kinds of works is shared to a plurality of computers constituting a computer group, the load can be shared uniformly.

Other objects, features and advantages of the invention will become apparent when reading the following detailed description of the embodiments of the invention in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram showing the schematic configuration of a parallel computer system in a load sharing control method according to an embodiment of the present invention;

FIG. 2 is a view showing an example of the resource utilization-state management table by work in the load sharing control method according to the embodiment of the present invention;

FIG. 3 is a view showing an example of the operation mode table in the load sharing control method according to the embodiment of the present invention;

FIG. 4 is a view showing an example of the load sharing grouping table in the load sharing control method according to the embodiment of the present invention;

FIG. 5 is a view showing an example of the parameter definition for defining resource utilization target values in the load sharing control method according to the embodiment of the present invention;

FIG. 6 is a flow chart showing a procedure of load sharing control in the load sharing control method according to the embodiment of the present invention; and

FIG. 7 is a flow chart showing a procedure of appropriate system selection based on the parameter definition in the load sharing control method according to the embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of the present invention will be described below in detail with reference to the accompanying drawings. In all the drawings for explaining the embodiment, parts having the same as or equivalent to each other are referenced correspondingly and the repetition of description is omitted.

A load sharing control method according to the present invention will be described below on the basis of an embodiment in which resources are assigned in accordance with the utilization target values of resources to be utilized in respective works.

FIG. 1 shows the schematic configuration of a parallel computer system in a load sharing control method according

to the embodiment. In FIG. 1, the reference numeral 1 designates a high-speed connection mechanism serving as an operation manager; 2 to 4, computers; 1a, 2a, 3a and 4a, central processing units; 1b, 2b, 3b and 4b, main storage devices (MS); 2c, 3c and 4c, operating systems; 2d, 3d and 4d, sub-systems; 2e and 3e, online processes; 2f, 3f and 4e, batch processes; 3g and 4f, TSS processes; 8a, a job input console; 9, a communication control processor (CCP) for controlling communication among the computers; 10, a communication network (such as LAN, WAN, etc.); and 11a to 11d, terminals which can be connected to the parallel computers through the network 10. Job input consoles 8a may be further provided in the computers 3 and 4 or may be provided in the terminals 11a to 11d.

As shown in FIG. 1, the parallel computer system in the load sharing control method according to the present embodiment comprises a high-speed connection mechanism 1, computers 2 to 4, central processing units 1a, 2a, 3a and 4a, main storage devices 1b, 2b, 3b and 4b, operating systems 2c, 3c and 4c, sub-systems 2d, 3d and 4d, online processes 2e and 3e, batch processes 2f, 3f and 4f, and TSS processes 3g and 4g. Each of the computers includes a scheduler for making a query to the operating system (OS) of the other computer, adjustment, etc. The main storage device 1b of the high-speed connection mechanism 1 has resource utilization-state management tables 1i (FIG. 2) and work target value setting table 1j (FIG. 2). Although this embodiment shows the case where the number of computers constituting a group of computers is three, it is a matter of course that the invention is not limited thereto and that any desired number of computers may be provided.

In the parallel computer system, a plurality of computers 2 to 4 constituting a group of computers are connected to the high-speed connection mechanism 1 which is a computer for managing the operations of the set of computers. The high-speed connection mechanism 1 and the computers 2 to 4 include central processing units 1a, 2a, 3a and 4a, and main storage devices 1b, 2b, 3b and 4b, respectively, so that operating systems (OS) 2c, 3c and 4c operate in the respective computers 2 to 4. Each of the computers monitors its own load state by referring to both the resource utilization-state management table and the work target value setting table from the high-speed connection mechanism and operates to make interaction about shared load through the CCP 9 on the basis of the result of the monitoring.

Among the computers 2 to 4 in the parallel computer system for carrying out the load sharing control method according to the present embodiment, a corresponding computer, in which its own scheduler for determining the computer to execute a work requested to be started operates, makes reference to the high-speed connection mechanism which stores therein the resource utilization-state management table 1i indicating the current utilization states of resources by work for the other computers and the work target value setting table 1j indicating the utilization target values of resources by work for the other computers.

In the computers 2 to 4 constituting the computer group in the parallel computer system for carrying out the load sharing control system according to the present embodiment, the sub-systems 2d, 3d and 4d for executing specific work processes respectively operate so that a set of the online process 2e and the batch process 2f, a set of the online process 3e, the batch process 3f and the TSS process 3g and a set of the batch process 4e and the TSS process 4f are executed in the sub-systems respectively in the form of load sharing.

FIG. 2 shows an example of resource utilization-state management tables by work in the load sharing control

method according to the present embodiment. As shown in FIG. 2, the resource utilization-state management tables 1*i* by work in the load sharing control method according to the embodiment are constituted by a resource utilization-state management table 5 for work A, a resource utilization-state management table 6 for work B, and a resource utilization-state management table 7 for work C.

Further, as shown in FIG. 2, in the resource utilization-state management tables by work in the load sharing control method according to the present embodiment, systems 1 to 3 on each of the resource utilization-state management tables 5 to 7 correspond to the respective computers 2 to 4, shown in FIG. 1, so that the resource utilization-state management tables 5 to 7 are generated by work or by group which will be described later.

The resource utilization-state management tables 5 to 7 in the load sharing control method according to the present embodiment store CPU utilization ratios which are processing periods per unit time of the central processing units 2*a* to 4*a* of the computers 2 to 4 in the case where works A to C are executed by the computers 2 to 4, main storage (MS) free ratios which are the proportions of free pages in the main storage devices 2*b* to 4*b*, and paging ratios indicating the proportions of the number of times of paging.

The central processing units 2*a*, 3*a* and 4*a* of the computers 2 to 4 constituting the group of computers in the parallel computer system for carrying out the load sharing control method according to the present embodiment have the same processing performance. When the group of computers are constituted respectively by computers having central processing units which are different in terms of processing performance, however, relative CPU utilization ratios may be used so that the processing performance of the computer having the lowest processing performance is regarded as '1'.

FIG. 3 shows an example of an operation mode table in the load sharing control method according to the present embodiment. In FIG. 3, the reference numeral 30 designates an operation mode table; and 31 to 33, operating processes corresponding to the respective systems 1 to 3.

As shown in FIG. 3, in the operation mode table in the load sharing control method according to the present embodiment, operation modes are formed from three kinds of work processes, namely, batch processes, online processes and TSS processes for the works 31 to 33. The system 1 corresponding to the computer 2 is assumed to be in an operation mode for executing a batch process and an online process. The system 2 corresponding to the computer 3 is assumed to be in an operation mode for executing all the processes, that is, a batch process, a TSS process and an online process. The system 3 corresponding to the computer 4 is assumed to be in an operation mode for executing a batch process and a TSS process.

FIG. 4 shows an example of a load sharing group definition table in the load sharing control method according to the present embodiment. In FIG. 4, the reference numeral 40 designates a load sharing group definition table; and 41 to 43, load sharing groups.

As shown in FIG. 4, the load sharing group definition table 40 in the load sharing control method according to the present embodiment has load sharing groups 41 to 43.

Further, as shown in FIG. 4, the load sharing group definition table in the load sharing control method according to the present embodiment indicates the definition of load sharing groups when a plurality of work processes are grouped so that the load sharing groups are made to be units

for load sharing. The group 1 is assumed to be a load sharing group 41 assigned for batch job processes A and B. The group 2 is assumed to be a load sharing group 42 assigned for online processes. The group 3 is assumed to be a load sharing group 43 assigned for TSS processes. When the online processes contain different works, they are grouped into different groups such as an accounting system and an information system in a banking system.

FIG. 5 shows an example of a parameter definition table for defining utilization target values of resources in the load sharing control method according to the present embodiment. In FIG. 5, the reference numerals 51 to 56 designate various kinds of parameters.

As shown in FIG. 5, in the parameter definition table for defining utilization target values of resources in the load sharing control method according to the present embodiment, grouping work processes for setting utilization target values of resources is defined by WLG (Work Load Group) parameters 51 to 53 in which WLG1, WLG2 and WLG3 define online process, batch process and TSS process respectively. Setting of the utilization target values of resources of works grouped by the WLG parameters is performed on the basis of WLR (Work Load Rate) parameters 54 to 56 in which, for example, the value of an upper load limit of each group is designated and is expressed in terms of percents of its CPU utilization.

In the parameter definition in the load sharing control method according to the present embodiment, an online process group is defined by the parameter 51 'WLG1 ONLINE', a batch process group is defined by the parameter 52 'WLG2 BATCH', and a TSS process group is defined by the parameter 53 'WLG3 TSS'.

Further, in the WLR parameters for parameter definition in the load sharing control method according to the present embodiment, the utilization target values of resources in process groups corresponding to 'WLG1', 'WLG2' and 'WLG3' in each system identified by 'SYSID' are expressed in terms of percentages.

For example, in the case of the parameter definition in the load sharing control method according to the present embodiment, as represented by the parameter 54 in FIG. 5, the resource utilization target values of the process groups 'WLG1' and 'WLG2' in the system 1 identified by 'SYSID=1', that is, the resource utilization target values of the online process group and the batch process groups are defined as '20%' and '80%' respectively.

The resource utilization target values in the case of parameter definition shown in FIG. 5 indicate the CPU utilization ratios of the respective computers. In the case of the parameter 54, the system 1 per se performs controlling so that online transaction is not given to the system 1 when the CPU utilization ratio indicating load on the online process is not smaller than '20%'.

FIG. 6 is a flow chart showing a procedure of load sharing control in the load sharing control method according to the present embodiment.

FIG. 6 shows an example of load sharing control in the load sharing control method according to the present embodiment. In a specific one of systems 1 to 3 corresponding to the computers, when a requested job is actually executed, the batch job load sharing job scheduler 60 for scheduling batch jobs to share load to the batch jobs issues a load sharing macro definition indicating inquiry of an OS supervisor (not shown) to its own OS (step 61). In the OS, the optimum system which is a destination of load sharing is determined by the OS supervisor (step 62). When the

judgment proves that the optimum system is its own system, the shared job is executed by its own system (step 63). When the judgment proves that the optimum system is another one, the other system is prompted to execute the job (steps 64 and 65).

In the step 62, the OS supervisor examines the current resource utilization states of computers constituting the computer group on the basis of management information from the high-speed connection mechanism 1, selects the optimum computer system from the computers on the basis of the parameter definition and informs the batch job load sharing job scheduler of the selected system.

The batch job load sharing job scheduler informed by the OS supervisor operates either its own computer system or another computer system correspondingly to the information from the OS supervisor so that the job requested to be started is executed by the selected optimum system.

The load sharing control is performed in the same manner as above also when the load of jobs on other groups is shared. For example, in load sharing control in online transaction, an online program issues load sharing macro definition in the step 61 so that the optimum computer system for the online process is selected by the OS supervisor.

Further, the load sharing control is performed in the same manner as above also for load sharing control for the TSS process. A TSS starting program issues load sharing macro definition in the step 61 and the macro definition is executed by the system designated by the OS supervisor in the step 62 to thereby achieve load sharing in accordance with the operation mode.

FIG. 7 is a flow chart showing a procedure of the optimum system selection based on the parameter definition in the load sharing control method according to the present embodiment.

As shown in FIG. 7, in the optimum system selection based on the parameter definition in the load sharing control method according to the present embodiment, the system selection process per se is called out in a load sharing macro procedure, so that the load sharing group definition table 40 is first searched in step 71 to examine which load sharing group contains the job to be subjected to load sharing.

Then, in step 72, the resource utilization-state management table is examined about the systems 1 to 3 corresponding to the computers 2 to 4 constituting the computer group so that the systems are arranged in the order of ascending the resource utilization ratio of the system as a whole.

In step 73, the resource utilizing state of the load sharing group containing the job retrieved in the step 71 is compared with the resource utilization target value based on the parameter definition in a manner so that the comparison is performed upon the systems arranged in the step 72 in the order of ascending the resource utilization ratio of the system as a whole.

When the result of comparison between the resource utilizing state of the load sharing group containing the job and the resource utilization target value based on the parameter definition in the step 73 proves that the resource utilizing state of the load sharing group containing the job is not smaller than the resource utilization target value based on the parameter definition, the procedure goes to step 74 in which the resource utilizing state of the load sharing group containing the job for another system is compared with the resource utilization target value based on the parameter definition by referring to the resource utilizing state of the load sharing group containing the job for the other system so

that the comparison is performed upon the systems arranged in step 72 in the ascending order of the resource utilization ratio of the system as a whole.

When the result of comparison between the resource utilizing state of the load sharing group containing the job and the resource utilization target value based on the parameter definition in the step 73 proves that there is some system in which the resource utilizing state of the load sharing group containing the job is smaller than the resource utilization target value based on the parameter definition, the procedure goes to step 75 in which a system in the resource utilization state smaller than the resource utilization target value based on the parameter definition is selected as a system for executing the job.

When the result of retrieval of the resource utilizing state of the load sharing group containing the job and the resource utilization target value based on the parameter definition with respect to the systems arranged in the step 72 in the ascending order of the resource utilization ratio of the system as a whole proves that there is no system in which the resource utilizing state of the load sharing group containing the job is smaller than the resource utilization target value based on the parameter definition, the lowest system in the resource utilization ratio as a whole is selected as a system for executing the job.

Although the present invention has been described above specifically on the basis of an embodiment, it is a matter of course that the invention is not limited to the embodiment and that various modifications or changes may be made without departing the gist of the invention.

What is claimed is:

1. In a parallel computer system having a computer group including a plurality of computers and an operation management mechanism which includes a computer for managing the operation of the computer group, a load sharing control method for sharing a load for executing a plurality of kinds of work processes among said plurality of computers in said computer group, said method comprising the steps of:

setting a resource utilization target value by each work for each of said plurality of computers included in said computer group in said operation management mechanism;

collecting a parameter value indicative of a resource utilization state, corresponding to each work, for each of said plurality of computers included in said computer group into said operation management mechanism;

selecting one computer to execute a newly requested work process from said plurality of computers included in said computer group on the basis of the differences between resource utilization target parameter values by respective works in said plurality of computers included in said computer group and parameters values indicating current resource utilization states corresponding to the respective works from said operation management mechanism; and

executing, in the selected computer, the newly requested work process.

2. A load sharing control method according to claim 1, wherein said parameter value indicating the resource utilization state corresponding to each work of said plurality of computers included in said computer group collected by said operation management mechanism includes at least one of CPU utilization ratio information for each of said computers included in said computer group, CPU utilization ratio information weighted by difference in performance of said

plurality of computers included in said computer group, and main storage available-state information in said plurality of computers included in said computer group.

3. A load sharing control method according to claim 1, wherein the step of selecting a computer to execute a newly requested work process includes a substep of calculating a difference between a resource utilization target parameter value corresponding to each work in each of said computers included in said computer group and the collected resource utilization state parameter corresponding to each work to thereby select a computer of which said difference is largest.

4. A load sharing control method according to claim 1, wherein said operation management mechanism includes a high-connection facility.

5. A load sharing control method according to claim 1, wherein said plurality of kinds of work processes includes at least two of an online process, a batch process and a time-sharing system process.

6. In a parallel computer system having a computer group including a plurality of computers and an operation management mechanism which is a computer for managing the operation of the computer group, a load sharing control apparatus for sharing a load for executing a plurality of kinds of work processes to said plurality of computers in said computer group comprises:

means provided in said operation management mechanism for setting a resource utilization target value corresponding to each work for each of said plurality of computers included in said computer group;

means provided in said operation management mechanism for collecting a parameter value indicative of resource utilization state corresponding to each work for each of said plurality of computers included in said computer group into said operation management mechanism;

means provided in each of said plurality of computers included in said computer group for selecting one computer to execute a newly requested work process from said plurality of computers included in said computer group on the basis of the differences between resource utilization target parameter values by respective works in said plurality of computers included in said computer group and current values of a parameter values indicating current resource utilization states by the respective works from said operation management mechanism; and

means for executing, in the selected computer, the newly requested work process.

7. A load sharing control apparatus according to claim 6, wherein said parameter value indicating the resource utilization state corresponding to each work of said plurality of computers included in said computer group collected by said operation management mechanism includes at least one of CPU utilization ratio information for each of said computers included in said computer group, CPU utilization ratio information weighted by difference in performance of said plurality of computers included in said computer group, and main storage available-state information in said plurality of computers included in said computer group.

8. A load sharing control apparatus according to claim 6, wherein said means for selecting a computer to execute a newly requested work process includes means for calculating a difference between a resource utilization target parameter value corresponding to each work in each of said computers included in said computer group and the informed resource utilization state parameter corresponding to each work to thereby select a computer of which said difference is largest.

9. A load sharing control apparatus according to claim 6, wherein said plurality of kinds of work processes includes at least two of an online process, a batch process and a time-sharing system process.

10. In a parallel computer system having a computer group including a plurality of computers and a shared storage unit for dynamically storing resource load information of the computer group which includes a computer for managing the operation of the computer group, a load sharing control method for sharing a load for executing a plurality of kinds of work processes among said plurality of computers in said computer group, said method comprising the steps of:

setting a resource utilization target value by each work for each of said plurality of computers included in said computer group in said operation management mechanism;

collecting a parameter value indicative of resource utilization states, corresponding to each work, for each of said plurality of computers included in said computer group into said operation management mechanism;

selecting one computer to execute a newly requested work process from said plurality of computers included in said computer group on the basis of the differences between resource utilization target parameter values by respective works in said plurality of computers included in said computer group and parameters values indicating current resource utilization states corresponding to the respective works from said operation management mechanism; and

executing, in the selected computer, the newly requested work process.

11. In a parallel computer system having a computer group including a plurality of computers and a shared storage unit for dynamically storing resource load information of the computer group which is a computer for managing the operation of the computer group, a load sharing control apparatus for sharing a load for executing a plurality of kinds of work processes to said plurality of computers in said computer group comprises:

means provided in said operation management mechanism for setting a resource utilization target value corresponding to each work for each of said plurality of computers included in said computer group;

means provided in said operation management mechanism for collecting a parameter value indicative of resource utilization state corresponding to each work for each of said plurality of computers included in said computer group into said operation management mechanism;

means provided in each of said plurality of computers included in said computer group for selecting one computer to execute a newly requested work process from said plurality of computers included in said computer group on the basis of the differences between resource utilization target parameter values by respective works in said plurality of computers included in said computer group and current values of a parameter values indicating current resource utilization states by the respective works from said operation management mechanism; and

means for executing, in the selected computer, the newly requested work process.

373

FIG. 1

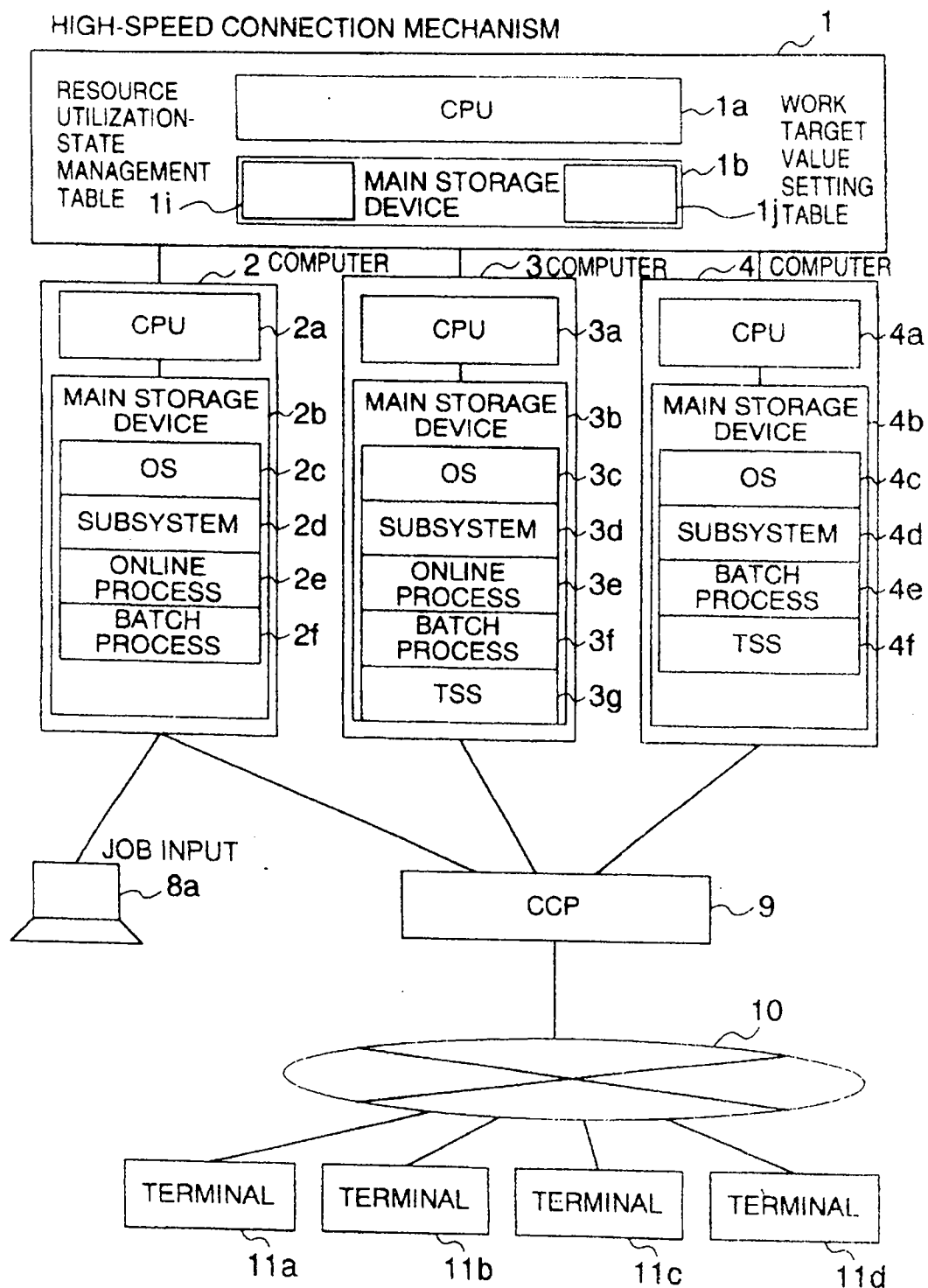


FIG. 2

WORK A

	CPU UTILIZATION	MS EMPTY RATIO	PAGING RATIO	...	~ 5
SYSTEM 1	80	60	70		
SYSTEM 2	75	35	50		
SYSTEM 3	20	20	30		

WORK B

	CPU UTILIZATION	MS EMPTY RATIO	PAGING RATIO	...	~ 6
SYSTEM 1	10	10	80		
SYSTEM 2	5	50	65		
SYSTEM 3	25	75	30		

WORK C

	CPU UTILIZATION	MS EMPTY RATIO	PAGING RATIO	...	~ 7
SYSTEM 1	45	100	0		
SYSTEM 2	45	80	0		
SYSTEM 3	45	0	0		

FIG. 3

	SYSTEM 1	SYSTEM 2	SYSTEM 3	~ 30 ~ 31 ~ 32 ~ 33
BATCH	←		→	
TSS		←	→	
ONLINE	←	→		
	ONLINE BATCH	ONLINE BATCH TSS	BATCH TSS	

375

FIG. 4

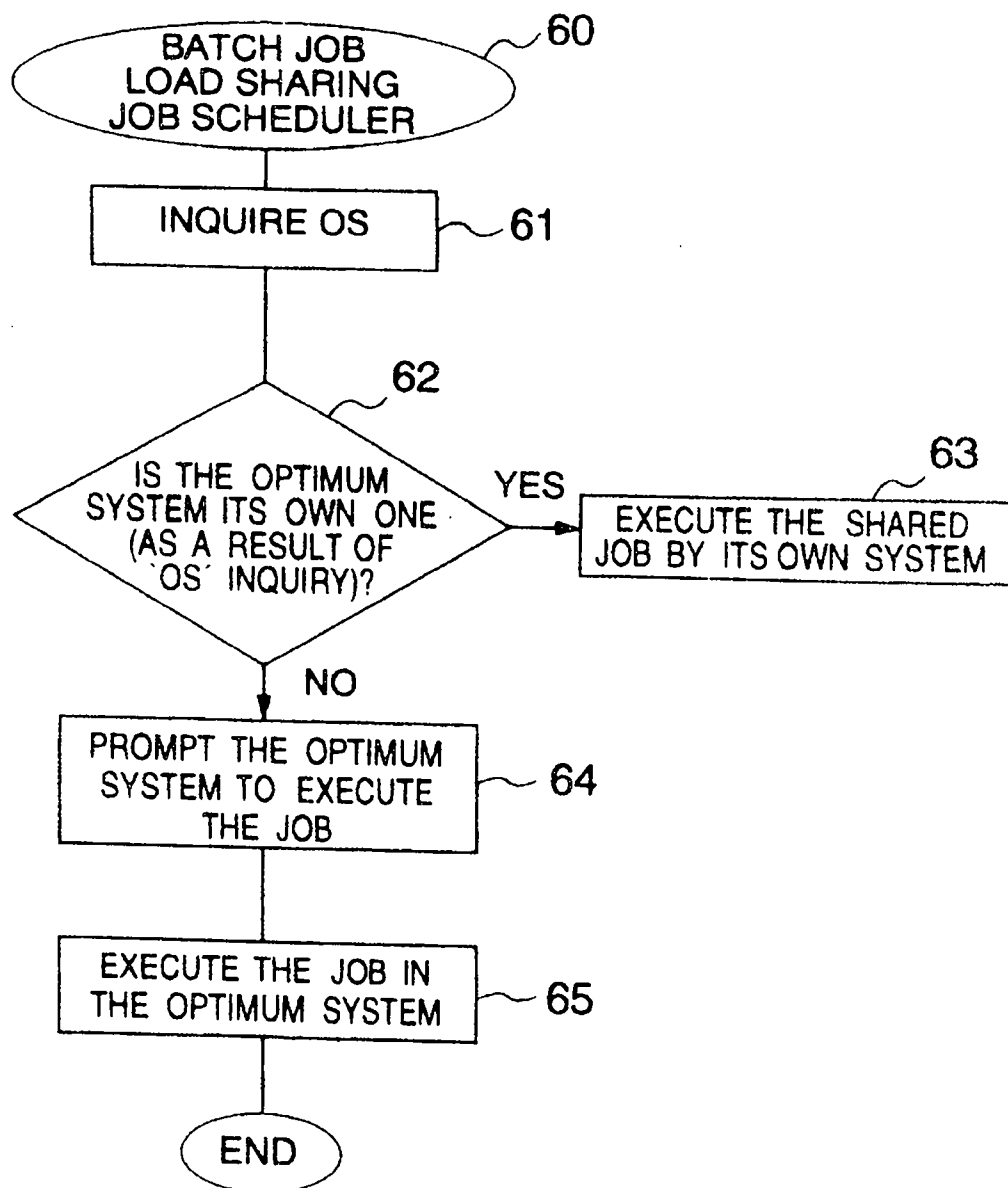
LOAD SHARING GROUP	PROCESS	40
GROUP 1	BATCH JOB PROCESS A, B	41
GROUP 2	ONLINE PROCESS	42
GROUP 3	TSS PROCESS	43

FIG. 5

WLG 1	ONLINE	51
WLG 2	BATCH	52
WLG 3	TSS	53
WLR	SYSID = 1, WLG 1 = 20, WLG 2 = 80	54
WLR	SYSID = 2, WLG 1 = 10, WLG 2 = 60, WLG 3 = 20	55
WLR	SYSID = 3, WLR 1 = 50, WLG 3 = 50	56

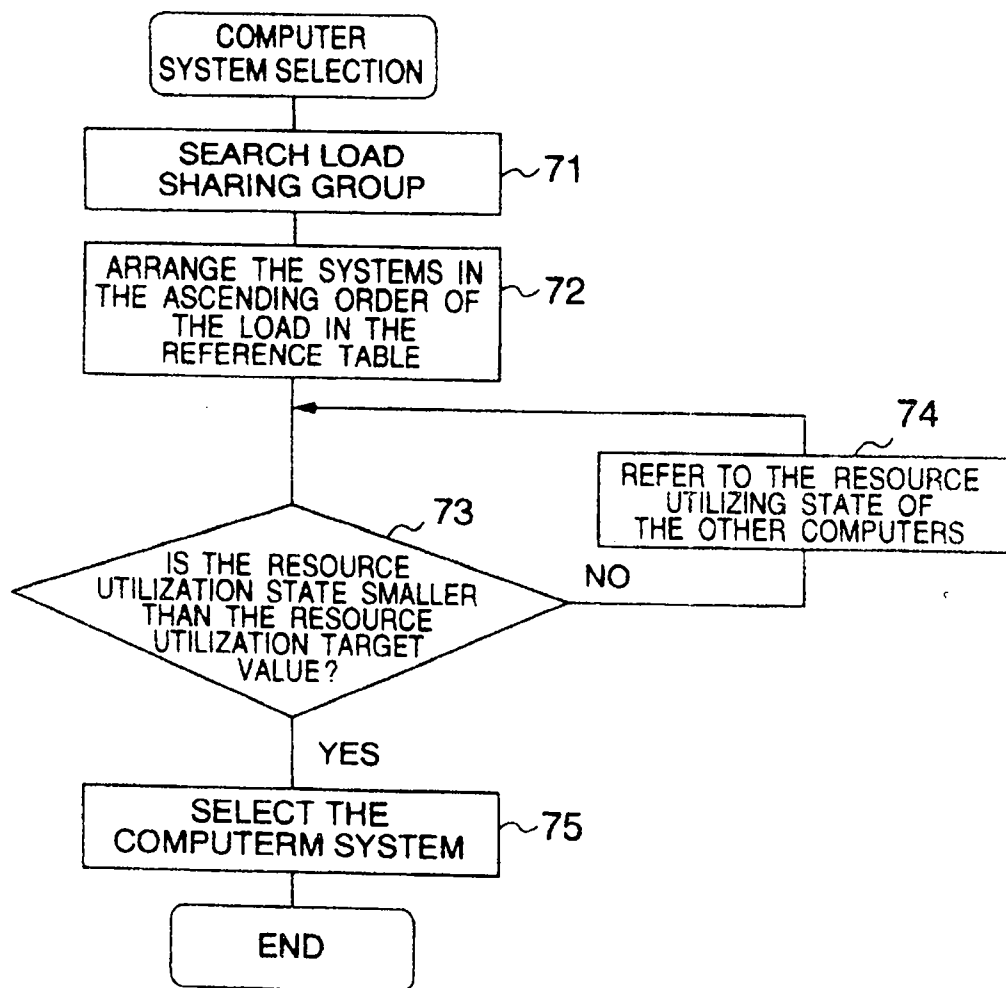
376

FIG. 6



377

FIG. 7





US006078955A

378

United States Patent [19]

Konno et al.

[11] Patent Number: 6,078,955

[45] Date of Patent: Jun. 20, 2000

[54] METHOD FOR CONTROLLING A
COMPUTER SYSTEM INCLUDING A
PLURALITY OF COMPUTERS AND A
NETWORK PROCESSED AS A USER
RESOURCE

5,881,238 3/1999 Aman et al. 709/226
5,884,037 3/1999 Aras et al. 709/226
5,886,643 3/1999 Diebhoft et al. 340/825,08
5,889,520 3/1999 Glaser 345/349
5,905,868 5/1999 Baghai et al. 709/224

[75] Inventors: Chisato Konno, Inagi; Katsuro
Kikuchi, Hachioji, both of Japan

Primary Examiner—Dung C. Dinh

Assistant Examiner—Abdullahi E. Salad

Attorney, Agent, or Firm—Antonelli, Ferry, Stout & Kraus,
LLP

[73] Assignee: Hitachi, Ltd., Tokyo, Japan

[21] Appl. No.: 08/917,726

[57] ABSTRACT

[22] Filed: Aug. 27, 1997

[30] Foreign Application Priority Data

Aug. 30, 1996 [JP] Japan 8-229693

[51] Int. Cl.⁷ G06F 13/00; G06F 15/16

[52] U.S. Cl. 709/224; 709/225; 709/226;
709/100

[58] Field of Search 709/224, 225,
709/226, 201, 202, 203, 223, 229, 100,
101, 102, 103, 104, 105, 106

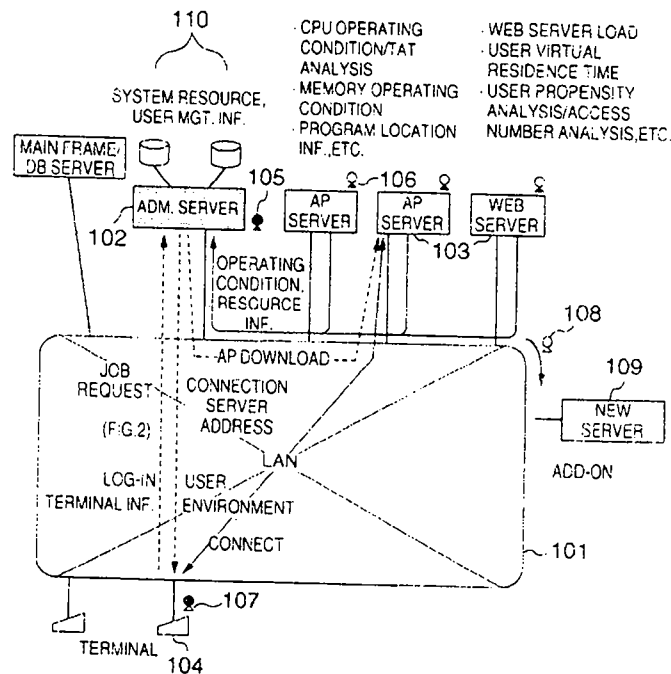
[56] References Cited

U.S. PATENT DOCUMENTS

5,341,477 8/1994 Pitkin et al. 709/226
5,867,706 2/1999 Martin et al. 709/105

In a computer system including a plurality of computers interconnected through a network, a specific computer A in the computer system manages the computer resource information and the operating condition information such as for a CPU, a file and a memory and the user resource information such as file data and a program of each computer of the computer system. When a user issues a request from a computer C to log in to the computer system, the request is reported to the computer A, and the information related to the user is returned to the computer C with reference to the user resource information held by the computer A. Then, the computer C acquires the file and the usable program held by the user in the computer system and executes a process, whereby the whole system can be used as resources of each user.

13 Claims, 18 Drawing Sheets



● SYSTEM ADM. AGENT

○ RESOURCE MGT. AGENT

○ SERVER MGT. AGENT

● TERMINAL MGT. AGENT

AP: APPLICATION PROGRAM

**METHOD FOR CONTROLLING A
COMPUTER SYSTEM INCLUDING A
PLURALITY OF COMPUTERS AND A
NETWORK PROCESSED AS A USER
RESOURCE**

BACKGROUND OF THE INVENTION

The present invention relates to a method of controlling a computer system including a plurality of computers interconnected by a network and used by a plurality of users through a plurality of types of computers, in which the whole computer system is usable as a computer resource of each user and provides an environment usable by each user without being conscious of the fact that a plurality of types of the computers are involved or the types of the computers included in the system.

In a conventional computer system including a plurality of computers interconnected by a network, an instruction (a process request) is given to the computer system by a user being conscious of the difference between the individual computers and the presence of files and programs to be used in the computer. A computer having a program is interrupted by a remote log-in or given an instruction to execute a program on the particular computer by a remote procedure call. Before execution of the program, however, it is necessary that a file required by the program be moved appropriately onto the computer.

In contrast, according to the CORBA (common object request broker architecture) specification proposed by the U.S. Object Management Group (OMG) assuring the coordination between the presence of a program for the whole computer system and the cooperation between programs, the use of and cooperation between the programs can be realized without being conscious of the actual presence of the programs on the computer system by registration in the ORB (object request broker) of the specification. This specification, however, defines a logical appearance of the presence of the programs registered in ORB in advance, in spite of the fact that the presence of the programs is fixed in the computers registered in advance. Also, there is no provision for the optimum use of the overall resources of the computer system.

Further, in a conventional computer system including a plurality of computers connected by a network, assume that a computer logging in to the computer system to execute some processing program has not any program or a file. Although such a program or a file can be retrieved from within the computer system through the network, the user is required to give an express instruction to the computer for retrieval in all cases.

Conventional computer systems intended to automatically select an optimum place of execution of a requested job from among the computers interconnected to a network are disclosed in JP-A-2-120968, JP-A-5-134959, JP-A-5-173990, etc. In these conventional systems, the place of executing a requested job is determined by monitoring the hardware resources such as the operating conditions and the amount of file of the computers connected to the network. Such a request is requested, since a plurality of computers are connected to the network. In this case, the location of a program and a file for executing a job are required to be indicated by the user consciously in advance.

Further, in a conventional computer system including a plurality of computers interconnected by a network, the interface is different in dependence on the computer (terminal) logging in to the computer system. In addition,

the type of the process that can be requested from a computer was also dependent on the particular computer.

SUMMARY OF THE INVENTION

5 An object of the present invention is to provide a method of controlling a computer system including a plurality of computers in which the files owned and the programs usable by each user on the computer system remain unchanged for the user, and regardless of the computer through which the computer system is logged in to, the use of the computer system by exactly the same interface is guaranteed.

10 Another object of the invention is to provide a method of controlling a computer system in which each user is not required to be conscious of the physical location of an intended program or an intended file in executing the process using the file or the program in the computer system.

15 Still another object of the invention is to provide a method of controlling a computer system in which the place of executing a process in the computer system is dynamically determined from the property and the load of the process requested and the prevailing utilization rate of the computer system.

20 According to one aspect of the present invention, there is provided a method of controlling a computer system including a plurality of computers, comprising the steps of causing a specific computer A of the computer system to manage the computer resource information and the operating condition information on the CPUs, the files and memories of each computer of the computer system and the user resource information such as file data and programs, reporting the computer resource information and the user resource information from each computer to the specific computer A from time to time, reporting first to the computer A any log-in request which may be issued by a user from a computer C, returning the information for the particular user from the user resource information held by the computer A, and causing the computer C to display a list of files held by the user and usable programs as an interface on the display of the computer C in the computer system, thereby making it possible to invariably manage the user resources in the system regardless of the computer through which the user logs in to the system.

25 According to another aspect of the invention, there is provided a method of controlling a computer system, including the steps of transmitting a program from a computer A, allowing the program to patrol the computers connected to the network and detect a computer newly connected to the network, and transplanting to the newly detected computer the program for reporting the computer resource information and the user resource information to the computer A, so that the newly connected computer can use the resources of the computer system as a new computer member of the computer system.

30 According to still another aspect of the invention, there is provided a method of controlling a computer system, including the steps of causing a computer C to issue a request for a process using a file and a program in the computer system, sending the request first to a computer A, causing the computer A to determine a computer or a computer group for executing the particular process on the basis of the physical location of the file and the program and the computer resource information, transferring and executing the file or the program, if required, on the basis of the particular determination, reporting to the requesting computer C the termination of the process, and causing the computer that has executed the process to send the file and the program

3

transferred thereto and any new file data generated as a result of the process to another computer, delete or report them to the computer A to the administration computer A of the computer system as a new user information resource of the computer, as required, thereby making it possible for the user to execute the requested process at an optimum place of execution without being conscious of the actual physical location of the file or the program.

According to this invention, there is provided a system including a plurality of computers interconnected through a network, in which the whole system can be used as a resource of each user. The data files and programs usable by each user in the system remain invariable regardless of a particular terminal from which the user logs in to the system, and the appearance (interface) is independent of the terminal. The user, therefore, is not required to use different methods for different terminals. Further, according to this invention, in the case where a user requests a process to the system including a terminal, the most efficient mode of executing the particular process is automatically determined by the system, and the resulting movement of the files and programs in the network is guaranteed by the system. All that is required of the user to do, therefore, is to request the process to a system. Also, since a file and a program are dynamically relocated taking the processing load of the whole system and the amount of movement in the network, an efficient utilization of the system is made possible. Further, in the case of any addition and connection of a new server, the file and the program thereof are automatically retrieved as a part of the system resources and are automatically allowed to be resident. The system can thus be expanded easily in keeping with the system load.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram for explaining the concept of the present invention.

FIG. 2 shows a flow of the overall process.

FIG. 3 shows a flow of the process for collecting the resource information of a server management agent program.

FIG. 4 shows a flow of the process for managing the server resource information of a system administration agent.

FIG. 5 is a diagram showing a structure of a system resource management table.

FIG. 6 shows a flow for registration of a user in the system.

FIG. 7 shows a flow of the process executed by a terminal management agent for logging in to the system.

FIG. 8 is a diagram showing a structure of a user management table.

FIG. 9 is an example of a user interface not dependent on the terminal.

FIG. 10 shows a flow for communication between the system administration agent and the terminal management agent.

FIG. 11 is a flowchart for determining a place of executing a job requested by a terminal.

FIG. 12 is a flowchart for determining a place of executing a job requested by a terminal.

FIG. 13 is a flowchart for determining a place of executing a job requested by a terminal.

FIG. 14 is a flowchart for requesting a process from a terminal management agent to a computer at the place of execution.

4

FIG. 15 is a diagram showing a structure of a file management table.

FIG. 16 is a flowchart for the operation of a resource management agent.

FIG. 17 is a flowchart for file transfer by a system administration agent.

FIG. 18 is a diagram showing a structure of an expanded file management table.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of the present invention will be explained below with reference to FIGS. 1 to 18.

First, explanation will be made of a concept and a system configuration intended to be achieved by the invention with reference to FIG. 1, followed by the description of a detailed flow of each processing section.

FIG. 1 shows an ordinary computer system including a plurality of computers interconnected by a network 101 to which the invention is applicable. The computers include at least a server shared by a multiplicity of users and a plurality of terminals 104 each occupied mainly by a single (not necessarily, specific) party. This system also includes an administration server 102 playing the central role in the invention. The administration server 102 is responsible for managing all the computer resources and all the jobs (the processing units of the computer) performed on the computers as well as the entire user information in the computer system.

First, the operation principle will be explained. All processing requests from the terminals 104 are provisionally sent automatically to the administration server 102. The administration server verifies the location of the program and the file corresponding to each processing request. Then, a computer (server or terminal) judged to be capable of handling a given process most efficiently is determined from the prevailing processing load of the servers in the network and the property and the location of the file and the program used for the intended process, and the place of execution of the process (connected server address) is notified to the particular terminal. In accordance with the information thus notified, the terminal requests a corresponding computer to execute the process. The movement of the file and the program, if required in advance, is the responsibility of the administration server. The user only requests the process, and the above-mentioned operations are automatically executed by the terminal, the administration server and the program (a unit of which is called an agent hereinafter) installed on the servers.

In order to realize this operation, the administration server 102 includes a system administration agent (program) 105, each server 103 includes a server management agent 106, and each terminal includes a terminal management agent 107, as a program. Also, a resource management agent 108 is available for transmission from the administration server.

The system administration agent 105 includes user management information, system resource management information and available application program information 110. The operating conditions and the resource management information of each server are received from the server management agent 106 to update the above-mentioned information. At the same time, the system administration agent 105 determines a computer and moves the file and the program in response to a process request sent from a terminal.

The server management agent 106 detects the resource management information (including the hardware resources such as the CPU, files and memories, and the software resources such as the user file data and the programs) of a server associated therewith, and transmits the information to the administration server 102. An arrangement is made to perform this detecting operation with the start or termination of a given job on the associated server as a motive.

The terminal management agent 107 detects the user information and the terminal information, and sends them to the administration server 102. Also, the terminal management agent 107 sends a processing request from the user to the administration server 102, and in compliance with an indication by the system administration agent 105 of a connected server address constituting the place of executing the particular process, connects the server and executes the process.

When a new server 109 is connected to the network, the resource management agent 108 is sent from the administration server 102 to the newly connected server 109, and a server management agent 106 is installed on the newly connected server 109. The new server can thus be made available as a part of the network system.

Then, a detailed flow of the operation will be explained. FIG. 2 shows the overall flow of a more specific process and a detailed flow of the corresponding succeeding process.

First, a mechanism for collecting and managing the system resource information and the operating condition information of each server by the process of the system administration agent 105 and the server management agent 106 will be explained with reference to FIGS. 2 and 3 to 5.

In FIG. 2, the server management agent 106 resident in each of the servers 103 acquires the system resource information and the operating condition information of a particular server. These information 205 are sent to the administration server 102. The information thus sent are stored in a system resource management information file 203 by the system administration agent 105.

FIG. 3 shows an operation flow of a server management agent program 106. The server management agent starts the operation described below (301) in response to a timer interruption from an OS, a process start interruption or a process termination interruption by a program. First, the change information on the file and the program resource is collected (302). This can be realized by using the date and hour information generated or updated by an OS instruction. Then, the capability and the capacity of the hardware resources (CPU, memory and file) are detected only at the first time of detection of the server resource (303). This is static information, for which the first one detection session is sufficient, and can be realized also using the OS function. As the next step, the utilization information of the hardware resource is collected also using the OS function (304). The collected information will be described in detail later with reference to FIG. 5. The collected information is sent to the system administration server (305). The data 306 thus sent is processed as shown in FIG. 4.

FIG. 4 shows a flow of the process for acquisition and management of the information 401 collected by the server management agent 106 in the system administration server 102. FIG. 5 shows a list of the information collected and managed and a storage file format. The system administration agent 105 acquires the information sent from the server management agent 106 (402). In accordance with the information thus acquired, the contents of the system resource management information file (FIG. 5) are updated (403).

The header 500 of the system resource management information table 203 shown in FIG. 5 has such information as a name for specifying an individual server computer, a CPU-ID for specifying a computer, and an IP (internet protocol) address in the system network. The header is linked by a pointer to a table having more detailed information. This pointer indicates a pointer table 501 having other pointers indicating a plurality of subtables having classified storage information. The subtables include a hard resource table 502 indicating the resources on the computer hardware, an operating condition table 505 for recording the operating conditions of the hard resources at each time point, and a file management table 510 for recording the information stored in the files on the server. Other information 520 can also be managed by a subtable having the same form linked from this pointer table whenever required.

The hard resource table 502 contains basically static information in the absence of any change or addition of the hardware which is collected in step 303 in FIG. 3. The operating condition table 505 is for recording the operating conditions of the hardware at the time of the interruption of step 301 in FIG. 3 and collected with the same interruption as a motive. This operating condition table 505 has stored therein the date and hour of collection 506, the CPU utilization 507, the memory utilization 508 and the file utilization 509. The file management table 510 has stored therein a file name 511, a location 512 such as the directory name and the folder name, a type 513 indicating a file data or a program, an attribute 514 indicating an interactive execution (I) or a batch-type execution (B) for a program, a user ID 515 indicating the holder of the file, an access right 516 indicating the scope of utilization of file copies and by other users, and a load amount 517 indicating a file capacity or, in the case of a program, a CPU power and a memory required for executing the program, with respect to the information stored in the file of a particular server.

As a result, the hardware resources and the software resources owned by each server are managed and so are the operating condition information at each time point. The data 306 sent from each server management agent shown in FIG. 3 can have the same format as this table or can be in a format compressed to contain only the sampled and updated information.

The embodiment of FIG. 3 shows the case in which the information is collected in the same manner for the timer interruption, the process start interruption and the process termination interruption. The sampled data, however, can be specified for one of the subtables of FIG. 5 in accordance with the type of interruption. As to the timer interruption, for example, a modification can be employed in which only the information corresponding to the operating condition table 505 is sampled.

Returning to FIG. 2, a method of realizing the user operability independent of the log-in terminal or the registration of the user management information by the system administration agent 105 and the terminal management information 107 will be explained with reference to FIGS. 2 and 6 to 9.

In FIG. 2, when the user logs in to the system, the terminal management agent 107 sends to the administration server 102 the log-in information and the terminal information 206 used for the log-in. The system administration agent 105 on the administration server 102 accesses the user management information 202 (FIG. 8) to process the log-in. At the same time, the user information 207 constituting a list of available programs and the file information owned by the user for the

entire network system is returned to the terminal management agent 107. The terminal management agent 107, using the particular information, displays the user interface (FIG. 9) remaining unchanged regardless of the terminal from which to log in and thus supports the user operation.

FIG. 6 is a flowchart showing the operation performed by a user for registering in the system for the first time. First, the user registration information 602 is entered by way of the terminal (601). This process can be normally operated only from a specific user (system manager or the like) or a specific terminal determined by the system. The system administration agent 105 sets an account and a password and registers the user resources such as a file in the user management file (FIG. 8) on the basis of the user registration information (604). After that, the user can log in to the system by the process shown in FIG. 7 in the capacity of a system user.

FIG. 7 shows a flowchart for the log-in operation performed by a user through the terminal management agent 107. First, an account and a password are input as the user log-in information (701). Then, the terminal management agent program extracts the terminal information used for the log-in (702). The terminal information is stored in a log-in terminal management table 809 of FIG. 8. The extracted log-in information and the terminal information 704 are sent to the system administration agent 105 of the system administration server (703).

The contents of the table of the user management information file (202 in FIG. 2) will be explained with reference to FIG. 8. The user management information table is constructed as a hierarchy table having a header (800) constituting a list of individual user IDs and corresponding passwords, and is linked further to a detailed information table by a pointer. This pointer indicates a pointer table 801 having a plurality of pointers for further indicating a plurality of subtables containing classified storage information. Each of the subtables includes a user management table 802 describing the right of the user to use the system, the charge rate, the total amount and upper limit of the charge, a user resource file management table 803 indicating a list of files owned or usable by the user in the whole internal area of the network system, and a log-in terminal management table 809 containing information associated with the terminal used by the user for logging in to the system. Other information is also managed in a subtable 820 of the same type linked from this pointer table. The user resource file management table contains only the information accessible by the user extracted from the file management table 510 of FIG. 5. The information included are limited to a file name 804, a location 805, a type 806 and an attribute 807. The log-in terminal management table 809 contains a log-in terminal IP address, a CPU performance representing the capability as a terminal computer, a memory capacity (absolute capacity and actual amount used), a file capacity, a display capacity (screen size, display colors, etc.), and a pointer indicating a resource file table locally owned by the terminal. The terminal resource file management table 810 holds the information of the same type as the file management table 510 shown in FIG. 5.

Returning to FIG. 7, the log-in information is checked using the tables of FIG. 8 on the basis of the information sent from the terminal, and the terminal information is registered (specifically, the tables 801, 809 and 810 shown in FIG. 8 are generated) (705). Though not shown here, any error or shortage of the registered information is reported to the terminal, in which case the subsequent operation is suspended. Then, a file (including a program) owned by the user

or the file available for use by the user is extracted (706). This operation is performed following the header 800 of the user management file and the pointer table 801 shown in FIG. 8 and referencing the table 803. As the next step, the user information thus extracted is sent to the terminal (707). Using the user information 708 thus sent thereto, the terminal management agent constructs and displays a user interface not dependent on the terminal in use (709).

An embodiment of a user interface assumed to be independent of a terminal is shown in FIG. 9. The conventional user interface is dependent to a large measure on the terminal used in that the information is displayed uniquely dependent on the window of the operating system installed for each terminal and in that the user resources existing in the network system other than the particular terminal are not displayed or, if displayed, in a different window using a specific command or the like. The interface according to the present invention, on the other hand, has a feature in that the user resources for the entire network system including a terminal are displayed independently of the particular terminal. All the files owned by the user and all the programs available for use by the user in the network system can be extracted by reference to the user management information table (FIG. 8) described above, and are sent to the terminal management agent from the system administration agent as the user information 708 shown in FIG. 7. A file locally owned by the terminal, which can also be referenced to the table 810 in FIG. 8, is originally sent from the terminal and located at the terminal. As a result, all the information on the files and the programs available for use when a process is executed in the system using a terminal is held by the terminal management agent. A user giving an instruction to the computer designates a file (data) used, a program constituting a unit of processing and a procedure indicating the sequence of processing. These information are classified into a file 901, a program (application program) 902 and procedure 903 and displayed by a common display means (such as icons) as shown in FIG. 9. However, the user is not required to be conscious of the physical location of the file, and therefore the information is displayed in a logical name attached by the user. The dashed lines of FIG. 9, therefore, are not displayed. For the user desirous of being conscious of the presence of the file, however, the hierarchical positions of the terminal (local), the server and main frame (back-end server) are specified in the windows as indicated by 905, 906, 907, so that an icon of a corresponding file is displayed within each area defined by the dashed lines. This is also the case with the program. A procedure, on the other hand, can be displayed as indicated by 908 in a subwindow by icons making up tools capable of describing the processing steps by combination of a file and a program, which icons are appropriately selected for prompting a detailed description. The user designates a process using this interface. As seen from the foregoing description, no information unique to a terminal or the system is used at all for constructing a screen. No matter which terminal the system is logged in from, therefore, a common user interface can be displayed. Now, explanation will be made about how the process designated by using this interface is interpreted by the terminal management agent and how to realize an actual process.

Returning to FIG. 2 again, a method of determining the place of executing the process at the administration management server 102 in accordance with a request made from the terminal through the terminal management agent 107, constituting another process of the system administration agent 105 and the terminal management agent 107, and a

382

method of executing the process based on such a determination, will be explained with reference to FIGS. 2 and 10 to 15.

First, a flow of the process will be briefly described with reference to FIG. 2. When a job request 208 is sent from a terminal through the above-mentioned interface to the administration server 102, the system administration agent 105 detects the physical location of the file and the program using the information contained in the user management information table 202 and the system resource management information table 203. Further, in executing the process, a computer (server or terminal) considered the most appropriate for the whole system is determined, and a connection server address is sent to the terminal accordingly. Also, if required, the file and the program are loaded by the system administration agent 105 (211, 212). The terminal management agent 107 requests the process to the connection server (including the terminal). Actually, however, the process, which is executed in one of the forms 201 shown in FIG. 2, is transparent to the user.

Now, a flow of operation for determining the place of executing a process will be explained with reference to FIG. 10. The job indication information 1001 input from the user through the above-mentioned interface is interpreted into an actual file name or an actual program name by the terminal management agent 107 (1002), and transmitted as a job request to the system administration server (1003). The system administration agent performs the following process 1005 based on the job request information 1004. The location of the file/program corresponding to the requested job is extracted using the user management information table 202. The place of execution of the particular job is determined with reference to the system resource management information table 203. At the same time, as required from the environment of the place of execution, the program and the file are loaded into the computer at the place of execution. The flow of operation of the process 1005 will be explained in detail with reference to FIGS. 11 to 13.

In the flowchart of FIG. 11, first, the location of the program/file of the requested job is detected from the user management information table 202, and using this location, the CPU processing performance C_i , the memory capacity M_i and the file capacity F_i are extracted as the resource information of the location with reference to the system resource management information table 203 (1100). This can be accomplished by referring to the field 805 describing the location in FIG. 8 and then the field 510 in FIG. 5 accordingly. Assuming that there are a plurality of locations, a prefix i is used. Then, the system administration agent 105 extracts a program size (P) indicating the memory capacity required for execution, a processing mode indicating the batch-type process B or the interactive process I , and an average processing load (L) for the program to be executed. Then, the data file amount (D) for a file to be processed is extracted (1101). If a program/file on the server is involved, the fields 513, 517 in the table 510 of FIG. 5 are referred to, while in the case of a terminal user resource, the fields 813, 817 in the table 810 of FIG. 8 are referred to. Then, the log-in terminal resource information including the CPU processing performance C_0 , the memory capacity M_0 and the file capacity F_0 are extracted from the user management information table 202 (FIG. 8) (1102). This can be accomplished by reference to the table 809 in FIG. 8. The processing load is assumed to be the total processing amount, and the CPU processing performance as the processing amount per unit time.

The place of job execution is determined in accordance with the following algorithm using these information.

First, two modes are involved depending on the possibility of moving the program (1103). In the case where a program execution agreement specifies (fixes) a computer to be used for execution, the program cannot be moved.

In the case where the movement is prohibited, the possibility of execution at a log-in terminal is checked in the presence of the program on the log-in terminal (1104). If the file to be processed is existent on the terminal (1105), the terminal is determined as the place of execution and the file need not be moved (1106). In the case where there is no file to be processed on the terminal (where the judgement in 1105 is NO), on the other hand, the possibility of moving the processing file to the terminal is checked (judged in step 1107 on the basis of whether the capacity D of the file to be processed is smaller than the file capacity F_0 of the terminal). If the movement is possible, the file is moved with the place of execution as a terminal (1108). In the case where the movement is impossible (in the case where the judgement in step 1107 is NO) or in the absence of a program on the terminal (in the case where the judgement in step 1104 is NO), on the other hand, the process proceeds to the next step for studying the execution of the program at another location.

Until a server satisfying the conditions is found by tracking the servers (i) having a program in the ascending order of the processing load, the process described below is repeated (1109). The information on the working load for the server having the particular program can be determined by referring to the operating condition table 505 of the system resource management information table 203 in FIG. 5. The servers can be selected, for example, in the descending order of the absolute value of the margin of the CPU processing capacity. The margin can be determined by subtracting the latest value of 507 from the absolute capacity of 502. Each server to be processed is checked as to whether it has a sufficient processing capacity (1110). To assure a sufficient processing capacity, the conditions described below are imposed on the CPU capacity and the memory capacity according to this embodiment. Specifically, it is determined whether the CPU capacity C_i is not less than a certain level ($L < \alpha \cdot C_i$) for an assumed processing load L , whether the memory capacity M_i on the server is not less than a certain proportion ($P < \beta \cdot M_i$) of the memory size P required for program execution, and whether the memory capacity including the file is sufficient ($P < (M_i + F_i)$ assuming that the file is used for virtual storage, where α and β are given constants. These values, however, are determined in such a manner that the processing time is not extremely lengthened. In view of this, α and β are set to 10 and 0.5, respectively, for example. Once these conditions are met (1110), whether the processing file can be stored on the server ($D < F_i$) is checked (1111). If such storage is possible, the particular server is determined as the place of execution (1112), and in the absence of the processing file on the server, the file is moved (copied) (1113). Otherwise, the next server candidate is checked.

In the case where the program can be moved, a priority place of execution (location of the terminal, or the program/file) is determined in accordance with the processing property and the amount of the program to be processed and the file capacity. An actual place of execution is thus determined while applying the conditions for judgement described above (1115). The flow of this operation will be described below with reference to FIG. 12.

The process for monitoring the movement of the program and file can be executed as a part of the process 1116 for registering the place of execution 1802 with reference to the table of FIG. 18.

11

In the case where the program processing mode is interactive, the terminal is determined as a place of execution in priority and whether the processing is possible or not is checked (1200). This is by reason of the policy of executing the processing at a place as near to the user as possible to maintain the performance of frequent interactive operations. In the case of interactive process, the possibility of execution on the log-in terminal is checked (1202 to 1205). This is accomplished with exactly the same algorithm as the process of 1110 to 1113. If execution on the log-in terminal is possible, the program and/or file is moved (copied) to the terminal as required. In the case where the execution on the log-in terminal is impossible due to the limitation of the CPU performance or the file capacity, in contrast, or in the case of batch-type processing, the steps described below are followed.

First, in the case where the absolute value of the processing load of an intended program is not more than a predetermined processing amount γ , for example, the possibility of execution is checked assuming the location of the program as a priority place of execution (1210). The processing amount γ is determined in advance, for example, as an assumed total processing amount of 10 billion instructions. This is in view of the policy of selecting a place of execution having a large processing capacity for a process larger than a predetermined level, while in the case of a process involving a smaller capacity, however, the possibility of execution at the program location is checked (1212 to 1215). This is accomplished by exactly the same algorithm as the process 1110 to 1113 in FIG. 11. In the case where this execution is possible, the program and file are moved (copied) to the server at the place of execution determined in the above-mentioned manner. In the case where there are a plurality of program locations, this process is repeated as many times as the locations until the place of execution is determined.

In the case where the data amount (D) of the file to be processed is a given size δ or more, on the other hand, the possibility of execution is checked with the file location as a priority candidate of the place of execution (1220). The given size δ is predetermined at, say, 3 Mbytes. This is by reason of the policy to avoid the movement of a large-scale data base or the like through the network. In the case where the data amount (D) of the file to be processed is not less than the given size δ (1221), on the other hand, the possibility of execution at the file location is checked (1222 to 1225). For this purpose, as in the preceding case, exactly the same algorithm as the steps 1110 to 1113 of FIG. 11 is used. If the execution is possible, the program is moved (copied) to the server at the place of execution determined, as required. In view of the fact that the file is already located on the server, however, the condition judgement 1223 for the file capacity is always YES. When there are a plurality of file locations, this process is repeated as many times as the number of such locations until the place of execution is determined.

In the case where the place of execution fails to be determined as a result of the judgement conducted so far, a place of execution most suitable from the viewpoint of the resources is determined from among all the servers included in the network.

In FIG. 13, the process described below is repeated (1301) until a server satisfying the conditions is found in the ascending order of the operating load among the servers (i) in the network. The information on the lightweight of the operating load, like in the process at 1109 of FIG. 11, can be known by reference to the operating condition table 505 of

12

the system resource management information table shown in FIG. 5. For example, the absolute values of the remainders of the CPU processing capacity are selected in the descending order, in which the remainder can be determined by subtracting the latest value of the CPU load 507 from the absolute capacity of 502. As an alternative, the latest average load amount can be used instead of the latest load. The possibility of execution is checked in the order of servers selected (1302 to 1305). This is accomplished by exactly the same algorithm as the steps 1110 to 1113 in FIG. 11 in the preceding case. In the case where the execution is possible, the program and file are moved (copied) to the server at the place of execution as required.

A detailed flow of operation performed by the system administration agent 105 was described above for determining the place of executing the job requested from the terminal.

Returning to FIG. 10, once the place of execution is determined by the above-mentioned algorithm, the place of execution and the information on the file moved are sent to the terminal, while such algorithm fails to be determined, on the other hand, the cause thereof is sent to the terminal (1007). As the information of 1008, the CPU-ID and the IP address are sent for the place of execution, and the CPU-ID and the IP address of the origin of the program or file is sent for file movement. The terminal management agent 107 that has acquired the transmission information at step 1010 requests a job to the computer at the place of execution on the basis of such information (1011), and communicates with the computer at the place of execution in accordance with the job contents. A detailed flow of the process performed by the terminal management agent will be explained below with reference to FIG. 14.

In FIG. 14, in the case where the place of execution fails to be determined (1401), the cause of failure to determine the place of execution is displayed on the screen waiting for the next instruction from the user (1402). The probable cause of failure to so determine is an insufficient resource at a specified server at the program location for an immovable program, or an insufficient resource of the system as a whole for a movable program.

As soon as the place of execution is determined, a job is requested to the computer in accordance with the mode and the place of program execution (1403). The destination of the requested job can be uniquely determined by the CPU-ID and the IP address. An actual job is executed in any one of the forms shown in 201 of FIG. 2, including the case in which the program on the client terminal is started (1404), the case in which the program on the server is interactively executed (1405), or in the case where the program on the server is executed in batch (1406). In execution, a communication is established with the computer at the place of execution in accordance with the particular job according to the program executed. The terminal management agent 107 participates at the termination of the job. At the time of job termination, a new file may be generated as a result of execution of the process, in which case the file is required to be registered as a user file. In the case where a new file is generated (1407), the system administration server registers the newly generated file (1408). The newly-generated file information 1409 is sent to the system administration agent 105, which in turn registers the file information in the file management table 510 of the related server of the system resource management information table 203 shown in FIG. 5, while at the same time registering the file information in the file management table of the user management information table shown in FIG. 8. Then, in the case where the

program has been loaded at the place of execution (1410), the program loaded at the place of execution (terminal or server) is depleted (1411) as the job termination process. In the case where the file was moved (copied) to the place of execution (1412) and in the case where the file copied to the place of execution (terminal or server) is updated, then the update information is reflected in the file at the origin and the file moved to the place of execution is deleted (1413). The information on whether the program is loaded or not or the information on the origin of movement are acquired by the terminal management agent 107 from the information of 1008 in FIG. 10.

An embodiment was described above, in which the whole system is managed integrally by the cooperation among the system administration agent 105, the server management agent 106 and the terminal management agent 107. In this method, regardless of the terminal from which a user logs in to the system, the resources (program and file data) usable by the user in the system including the appearance (interface) thereof remain unchanged for the user. Also, in response to a job request from a terminal, an optimum place of execution in the system is automatically detected and processed.

Several modifications of this embodiment will be explained below.

Although the present embodiment was described with a single job, the process of Executing several jobs can also be realized by the cooperation between the system administration agent and the terminal management agent in similar fashion. Two methods are available. In a first method, the place of execution is determined by the system administration agent using the above-mentioned algorithm for determining the place of execution repeatedly at the time of a request for all the jobs in coordination, and the resulting determination is returned to the terminal management agent which executes the jobs sequentially. In a second method, the system administration agent is inquired of the optimum place of execution of each job sequentially each time, which job is executed, followed by determining the place of execution of the succeeding job. In realizing this process, the process at the terminal management agent described above is repeated as many times as the number of jobs in coordination.

According to this embodiment, the processing load is held in the field of 516 in FIG. 5 as an estimated value (or a statistical value) of an average load of the program, and the place of execution is determined with reference to the estimated value. The processing load, however, depends also to a large measure on the file to be processed and is not an established value. An alternative method, therefore, is for the user to report (enter) the amount of the processing load at the time of job request, which amount can be used as a substitute.

According to this embodiment, the place of execution of a process is determined by a predetermined algorithm (FIGS. 11 to 13), and the place of execution is searched for at the terminal, the program location and the file location in that order of priority. In a modification, however, the file location can be considered in priority to the program location. Unlike in the present embodiment in which the order is fixed for searching for the place of execution by the system, the user may alternatively select the order.

According to this embodiment, the parameters α , β , γ used in the judgement equation for determining the place of execution are fixed in FIGS. 11 and 13. These parameters, however, may alternatively be changed dynamically in

accordance with the amount of the resources of the system as a whole. As another alternative, these parameters may be set in accordance with the distribution of average capacity of the servers making up the system.

According to the present embodiment, a program loaded at a given place of execution for a given process is deleted unconditionally after the termination of the process (1411, 1413 in FIG. 14). Instead, a program frequently loaded may remain loaded, and can be used as it is without being reloaded in the next session. This can be realized by registering the particular program in 510 of the system resource management information table 203 in FIG. 5. For registration, only the location on the corresponding line of the table of the loading origin of the program is changed and copied to the location on the server at the place of execution.

According to this embodiment, the process is realized based on the system resource management table of FIG. 5 with the computer as a header. In this embodiment, the location of a program is detected by searching the table of 510 of FIG. 5 from the computer. In contrast, a method is conceivable in which a table is prepared with a file as a header. FIG. 15 shows a configuration of such a table. This table is so configured that a plurality of locations of a file can be held. Numeral 1500 designates a table having the same contents as the table 510 in FIG. 5. In the table 1500, however, the location 1509 provides a pointer to the subtable 1510 indicating the location. The subtable 1510 includes a CPU-ID 1511 for specifying a computer, an IP address 1512 for indicating an address in the system network, and a directory 1513 and a folder 1514 indicating a position on the computer as a file location. Any alternative specific name which may exist on the individual computer is stored in an alias name 1515. In the presence of other locations, a plurality of locations can be stored by a pointer 1516 to another subtable of the same format. Using this table, the server having a program 1109 in FIG. 11 can be easily extracted. Also, this table can be referenced for other information as required, while the table 510 in FIG. 5 and the table 803 in FIG. 8 can be referenced as a table containing only the file names. In spite of this, the matching with the system resource management table 203 is required to be assured. In the case where the loading program described above is made to reside as it is, for example, the operation is necessary for coupling the server to a corresponding file column as a subtable of 1510.

Now, the process to be executed in the case where a new server is connected to the network system will be explained with reference to FIG. 16. In the case of a timer interruption or upon detection of a new resource addition, the system administration agent 105 performs the process described below and retrieves the new server as a part of the resources in the network system. The connection or disconnection of the new server is checked (1601), or in some cases, the resource management agent program 108 is sent to the IP address (1602). The servers in the system are registered in 500 of FIG. 5, and a server for which information after the pointer is not set is judged as a new server. The main function of the resource management agent program 108 is to install and start the server management agent program on the server. First, the software resources of the server are checked (1603). For this purpose, the type and version of the operating system (OS) mounted on the server are checked. On the basis of this study, the server management agent program suitable for the operating system of the server is sent to the system administration agent 105 (1604). In response to this, the system administration agent 105 sends the corresponding server management agent program

(1606). The resource management agent 108 installs and starts the server management agent program on the server thereby to terminate the process. Subsequently, as described with reference to FIG. 3, the information is collected by the server management agent with a given interruption as a motive, and the information thus collected is reported to the system administration server (1607). If the system administration agent has received the particular information for the first time, a system resource table is generated (501, 502, 505, 510 in FIG. 5) and the acquired information is set (1608). As a result, the new server is built in as a part of the resources of this system. In the process, the resource management agent program is prepared in a language (such as Java language) free of the platform in a manner operable with any server.

In the process up to this stage, the optimum place of execution and the temporary loading of the file in the computer at the place of execution were first determined as required, on the basis of the file location. Once the system is started, however, the problem may arise as to a more appropriate location of the file taking the efficient operation of the whole system into consideration on the basis of the loading frequency and the statistical information on the transfer rate of the file. In the embodiments described below, a method of copying or moving the file between servers in such a case will be described with reference to FIGS. 17 and 18.

First, the file management table described with reference to FIG. 15 is expanded and arranged to record the server on which each file has been used for processing. FIG. 18 shows an example of such an expansion. A field is prepared in which to store the number of executions 1801 providing the number of times a particular file is used for processing and also to store the pointer 1802 indicating a table storing each place of execution. The subtable 1810 for storing the number of executions at each place of execution has the CPU-ID and the IP address indicating the place of execution and the total number of times 1805 used for the process. The subtable is connected by as many pointers as different places of execution. Once the place of execution of a job is determined, the number of executions 1801 for the particular file is counted up. If the place of execution is already registered in the chain of the places of execution of 1802, the number of executions 1805 in the subtable is counted up. Otherwise, a new subtable is connected to the tail end of the chain, and the number of execution is set to one. This operation, if inserted in 1116 of FIG. 11, can hold the information.

In FIG. 17, the system administration agent 105 performs the process described below at regular intervals of time thereby to dynamically rearrange the files. Regular intervals may be one week or each time the number of job requests reaches 1000, for example. First, files frequently used and less frequently used are extracted (1701). The number of files extracted can be specifically determined in the descending order of frequency in advance. This extraction is possible by reference to the number of executions 1801 of FIG. 18. Then, several leading servers in terms of light and heavy weights of the server load are extracted (1702). This can be accomplished by calculating the average margin of each server from the statistical consumption based on the operating condition table 505 of the system resource management information table of FIG. 5 and the hardware resource 502. Then, servers high and low in the utilization rate of the file capacity can be extracted using the system resource management information table in similar fashion. The number of the leading servers to be extracted can be determined as, say, 5% of the total number of files or as, say, 10% of all

the servers. In accordance with these figures, the file is moved under certain conditions.

The conditions can be set as follows:

- (1) In the case where a given file is used with a high degree of frequency and often used by being loaded in other servers, then the file is provided for the servers liable to be often loaded. A server with a light resource load is provided with a copy.
- (2) If a plurality of files of the same type used with a low degree of frequency are arranged on the server, a part of them are deleted from the server.
- (3) A file used with a low degree of frequency is moved from a server having a large utilization rate of the file capacity to a server with a small utilization of the file capacity.

These conditions are actually fulfilled by following the procedure described below. In the description that follow, characters n1, n2, n3, n4 designate predetermined constants. The resources are not rearranged to an extreme extent based on the information at a given time, but in view of the fact that the situation may subsequently change considerably in dependence on the utilization rate of the user, the number is set to 1 or 2 assuming that the rate of change is small.

The extracted file used with a high degree of frequency is subjected to the following process (1704). In the case where the leading n1 servers at the place of execution other than the server with the particular file (1704) have an unoccupied file capacity, the particular file is copied to that server and the management table is updated accordingly (1706). In this case, the servers at the place of execution other than the server having the file are extracted in the descending order of the number of executions 1805, from among those having the CPU-ID of other than the location of the file in the connection table 1810 in FIG. 18. Also, the management table is updated in such a manner that the subtable 1510 of the server to which the file is copied is generated in the chain of the locations in FIG. 18, and the subtable 1510 thus generated is connected to the tail end of the chain. Further, the file is registered in the file management table for the corresponding server of the system resource management table shown in FIG. 5. Furthermore, the leading n2 servers with small load (1707) are processed in the same way as in 1706 (1708).

An extracted file used with a low degree of frequency is subjected to the following process (1710). In the case where such a file is located at a plurality of places (1711), the files are deleted from n3 locations not used or in the descending order of server capacity, and the management table is updated accordingly (1712). In the process, both the file location 1509 and the place of execution 1802 can be determined by reference to the table in FIG. 18. The locations not used, therefore, can be extracted. Also, by reference to the table of FIG. 5, the capacity of each server can be extracted.

An extracted server with a low utilization rate of file capacity is subjected to the following process (1715). The leading n4 files used with low frequency on the server (1716) are moved to a server used with low utilization rate of file capacity, and the management table is updated accordingly (1717). This process is substantially identical to that of 1706. A server with a low utilization rate of file capacity can be extracted by reference to the tables in FIG. 5 in similar manner.

The present invention is not limited to the above-mentioned embodiments, but includes various modifications without departing from the scope and spirit of the invention.

386

We claim:

1. A method of controlling a computer system including a plurality of computers interconnected through a network, said method comprising the steps of:

causing each of said computers to acquire computer resource information and operating condition information for resources including a CPU, a file and a memory, and user resource information including file data and a program, and to monitor said information; causing a specific computer A in said computer system to collect said computer resource information, said operating condition information and said user resource information from each of said computers and to manage said information;

causing said computer A to select a computer C in said computer system as an optimum place of execution of a given process using the information regarding the location of said file data and location of said program and the information on the operating conditions of said computer resources when a user requests said given process using said file data and said program from said computer system through a computer B; and

causing said computer C to execute said process requested by said user;

wherein the most efficient mode of execution as of the time of the process request is dynamically determined with the whole of said computer system as the user resources.

2. A method of controlling a computer system according to claim 1, wherein said step of causing said computer C to execute the requested process includes the step of causing said computer B that receives an indication of the place of execution from said computer A to request said computer C again to execute the process that was requested by said user, whereby said process requested by said user is executed without the user being aware of the place of execution thereof.

3. A method of controlling a computer system according to claim 1, wherein said step of causing said computer C to execute requested process includes the step of moving or copying the corresponding program or file data from the location thereof in the absence of said program or file data, on said computer C, wherein said process is executed without said user being aware of the actual place of execution of the process requested by said user.

4. A method of controlling a computer system according to claim 1, wherein said step of causing said computer C to execute the requested process includes the step of reflecting a change in the file data at the original location thereof when there exists no corresponding file data on said computer C and said file data is copied from the location thereof, resulting in said file data being changed by said process.

5. A method of controlling a computer system according to claim 3, wherein said step of causing said computer C to execute the requested process includes the steps of:

judging whether a corresponding program not existing on said computer C and moved or copied from the location thereof and processed is to remain or be deleted from said computer C after termination of said process; and deleting said program based on said judgement.

6. A method of controlling a computer system according to claim 3, further comprising the steps of:

reporting a new file data as a resource of the user on said computer C to said computer A, said new file data being generated as a result of execution of the process on said computer C; and

causing said computer A to update said computer resource information, wherein the user resources are managed for the whole computer system regardless of a specific computer on which said process is executed.

7. A method of controlling a computer system according to claim 1, wherein said step of selecting said computer C includes the steps of:

determining whether movement of a program is possible; determining a place of execution in dependence on a load including the amount of calculation and the memory capacity of the requested process at each of said determining steps on the basis of four choices including the location of said program, the location of said file data, a computer through which said user has logged in to the system and a computer having a high processing capacity usable as of the time of said process request.

8. A method for controlling a computer system including a plurality of computers interconnected through a network, said method comprising the steps of:

causing each of said computers to acquire computer resource information and operating condition information for resources including a CPU, a file and a memory and user resource information including file data and a program, and to monitor said information;

causing a specific computer A in said computer system to collect said computer resource information, said operating condition information and said user resource information from each of said computers and to manage said information; and

extracting file data owned by a user and a program usable by said user over the whole of said computer system and displaying said extracted file data and said extracted program on a computer B, when said user logs on to said computer system through said computer B, wherein a uniform user interface is provided regardless of any computer that has logged on to the computer system.

9. A method of controlling a computer system according to claim 8, wherein said step of displaying said extracting data file and said extracted program includes the step of displaying the file data and the programs of said user on said computer B regardless of the locations of said file data and said programs.

10. A method of controlling a computer system including a plurality of computers interconnected through a network, said method comprising the steps of:

causing each of said computers to acquire computer resource information and operating condition information for resources including a CPU, a file and a memory and user resource information including file data and a program, and to monitor said information;

causing a specific computer A in said computer system to collect said computer resource information, said operating condition information and said user resource information from each of said computers and to manage said information;

detecting a newly connected computer X by patrolling the computers connected to said network through said network;

setting a program to said computer X, by said program computer resource information and operating condition information for resources including a CPU, a file, a memory and user resource information including file data and a program are acquired and sent to said computer A;

automatically placing said computer X under the management of said computer system; and

387

19

causing said computer A to move said file data and said programs of said computer existing before a connection of said computer X automatically to said computer X in order to improve execution efficiency of said computer system as a whole.

11. A method of controlling a computer system including a plurality of computers interconnected through a net work, said method comprising the steps of:

causing each of said computers to acquire computer resource information and operating condition information for resources including a CPU, a file and a memory and user resource information including file data and a program, and to monitor said information;

causing a specific computer A in said computer system to collect said computer resource information, said operating condition information and said user resource information from each of said computers and to manage said information;

causing said computer A to select a given computer C as an optimum place of execution of a process in said computer system using the information regarding the location of file data and location of a program and the information regarding the operating conditions of said computer resource when a user requests said process using said file data and said program to said computer system through a given computer B;

causing said computer C to execute said process requested by said user;

monitoring the utilization of the file data and the program and the movement of said file data and said program between computers for each process; and

moving or copying a selected one of said file data and said program from the location thereof to another computer on the basis of the statistical data obtained as a result of said step of monitoring the utilization and the movement between the computers of said file data and said program;

20

wherein said file data and said program are moved or copied according to the utilization thereof during a predetermined period.

12. A method of controlling a computer system according to claim 11, wherein said step of moving or copying said file data and said program includes the steps of:

generating parameters including the utilization rate of each of said files during a predetermined period, the frequency and destination of movement of each of said files for utilization thereof and the utilization rate of the file region at an actual location thereof; and

determining the movement and deletion of a file on the basis of the values of said parameters.

13. A method of controlling a computer system including a plurality of computers interconnected through a network, said method comprising the steps of:

causing a specific computer A in said computer system to manage the computer resource information and the operating condition information for resources including a CPU, a file and a memory and user resource information including file data and a program of each computer of said computer system;

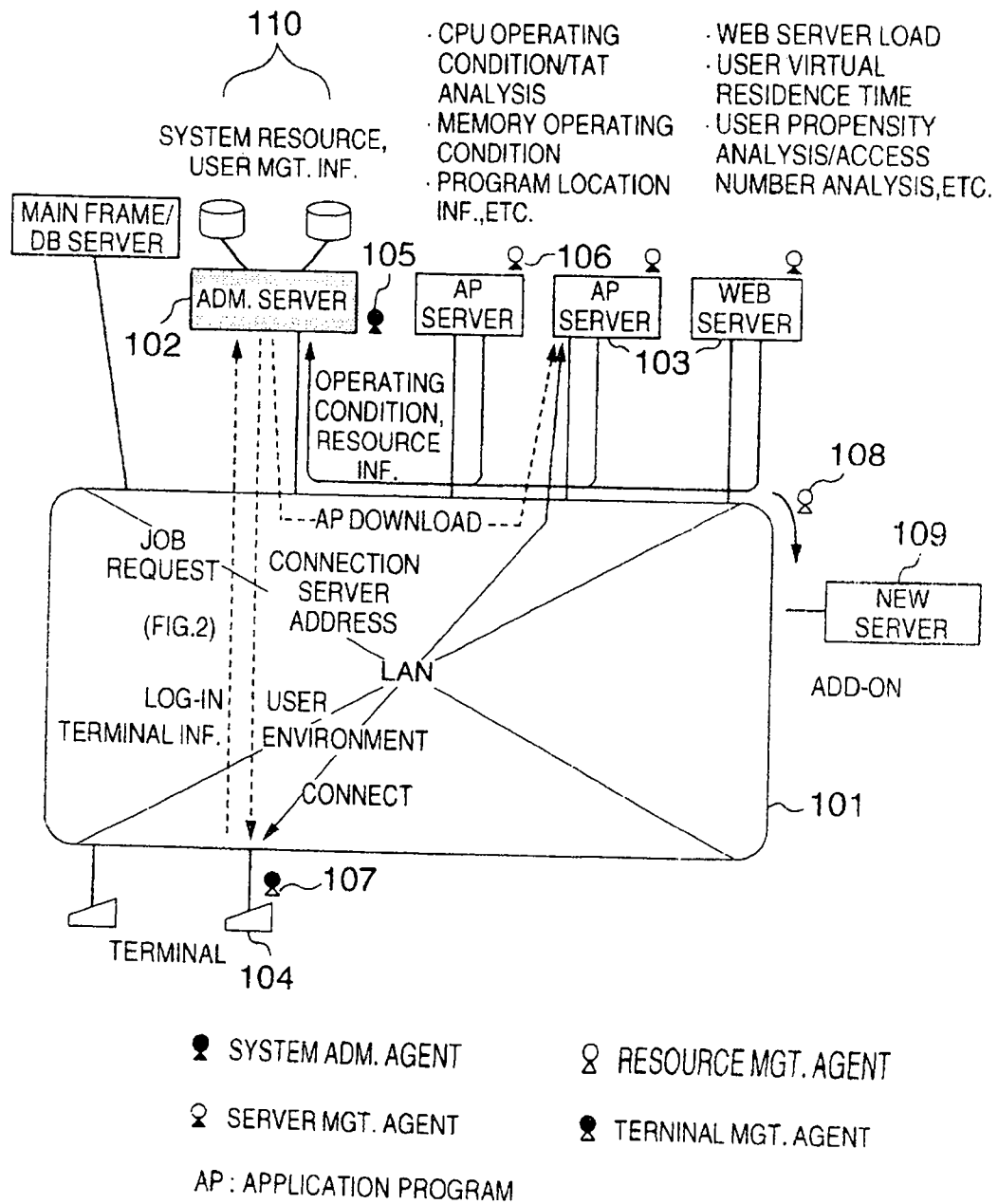
reporting any log on request issued by a user from a computer C in said computer system to said computer A, and returning the information on said user to said computer C with reference to the user resource information held by said computer A; and

causing said computer C to acquire a file and a usable program held by the user in said computer system and to execute a process using said information.

* * * * *

388

FIG. 1



390

FIG. 2

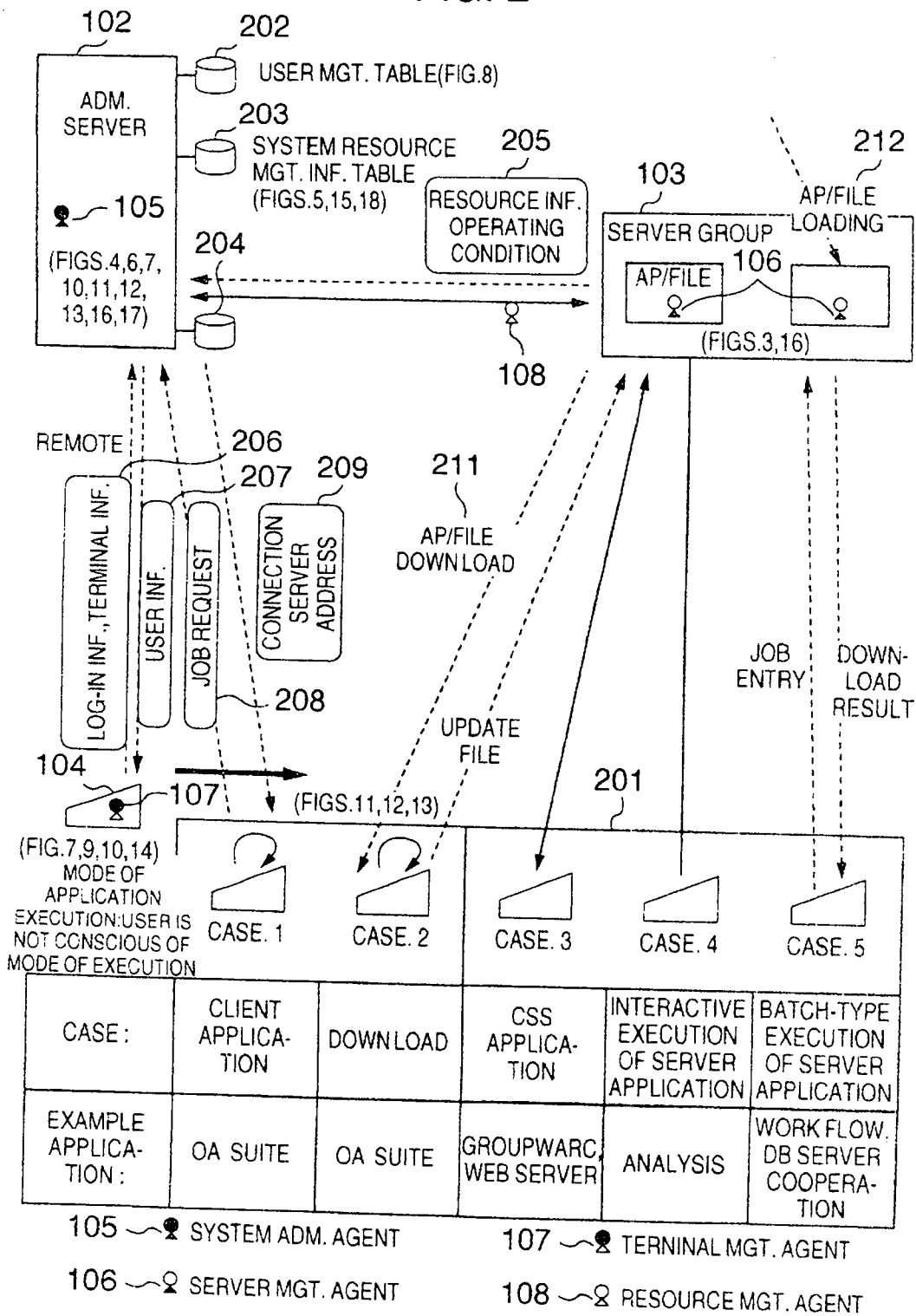
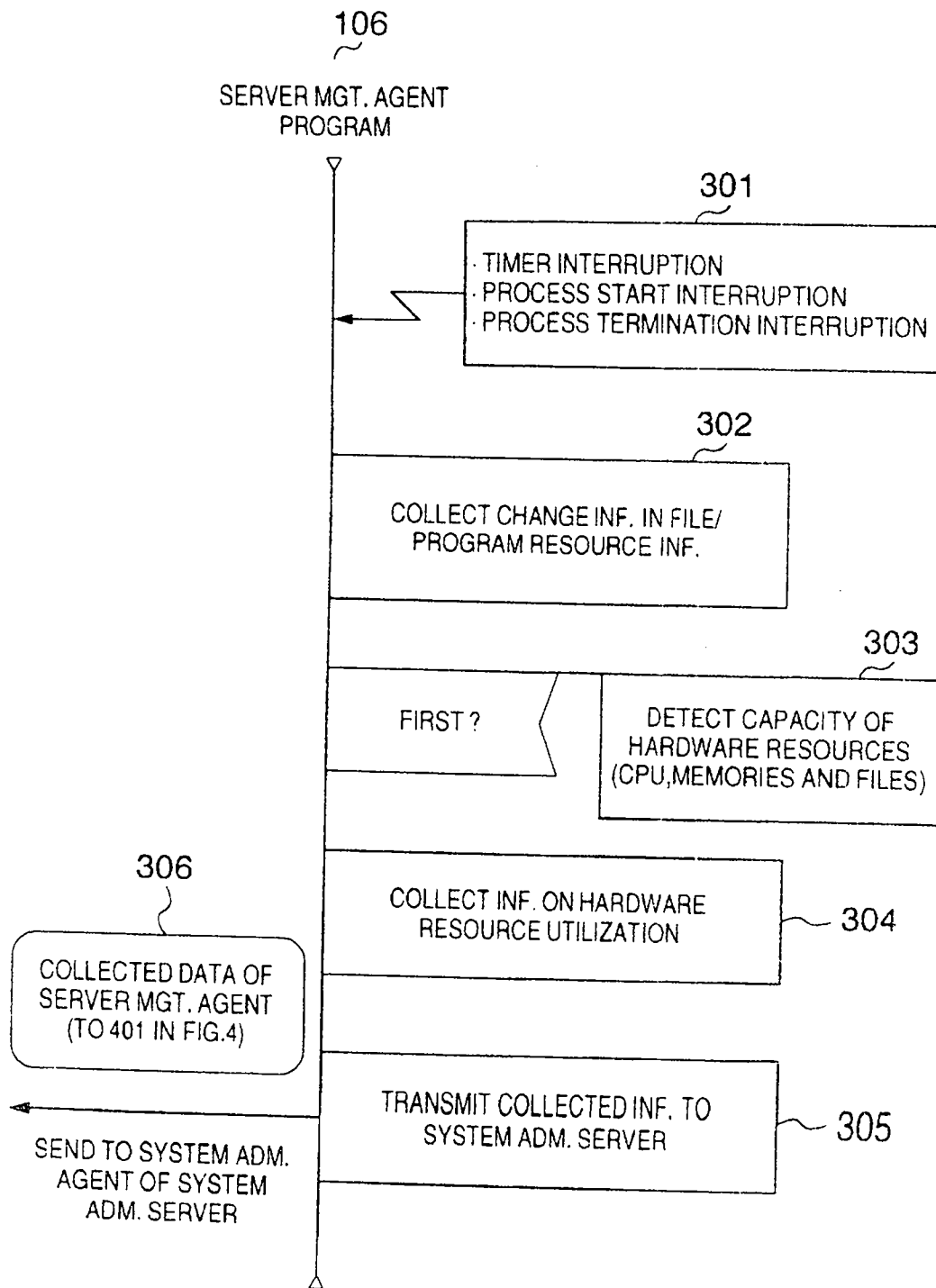
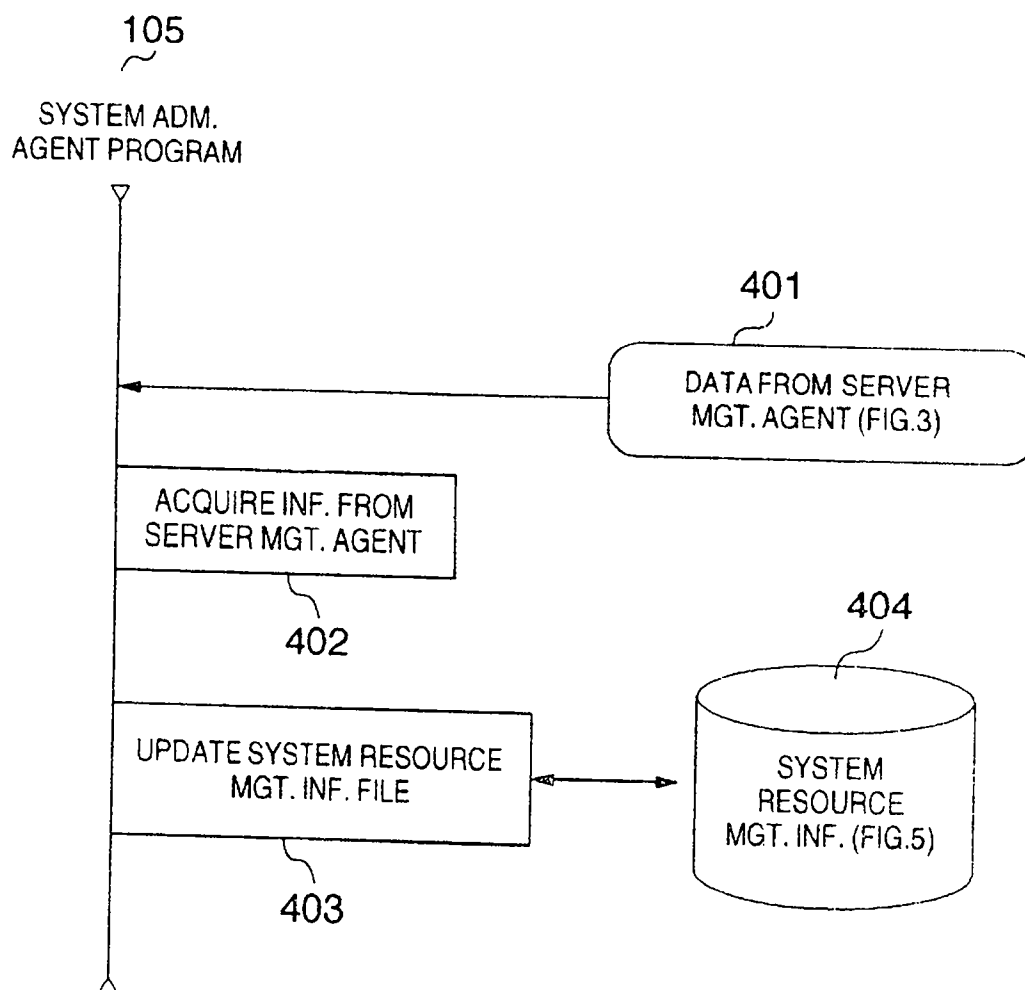


FIG. 3



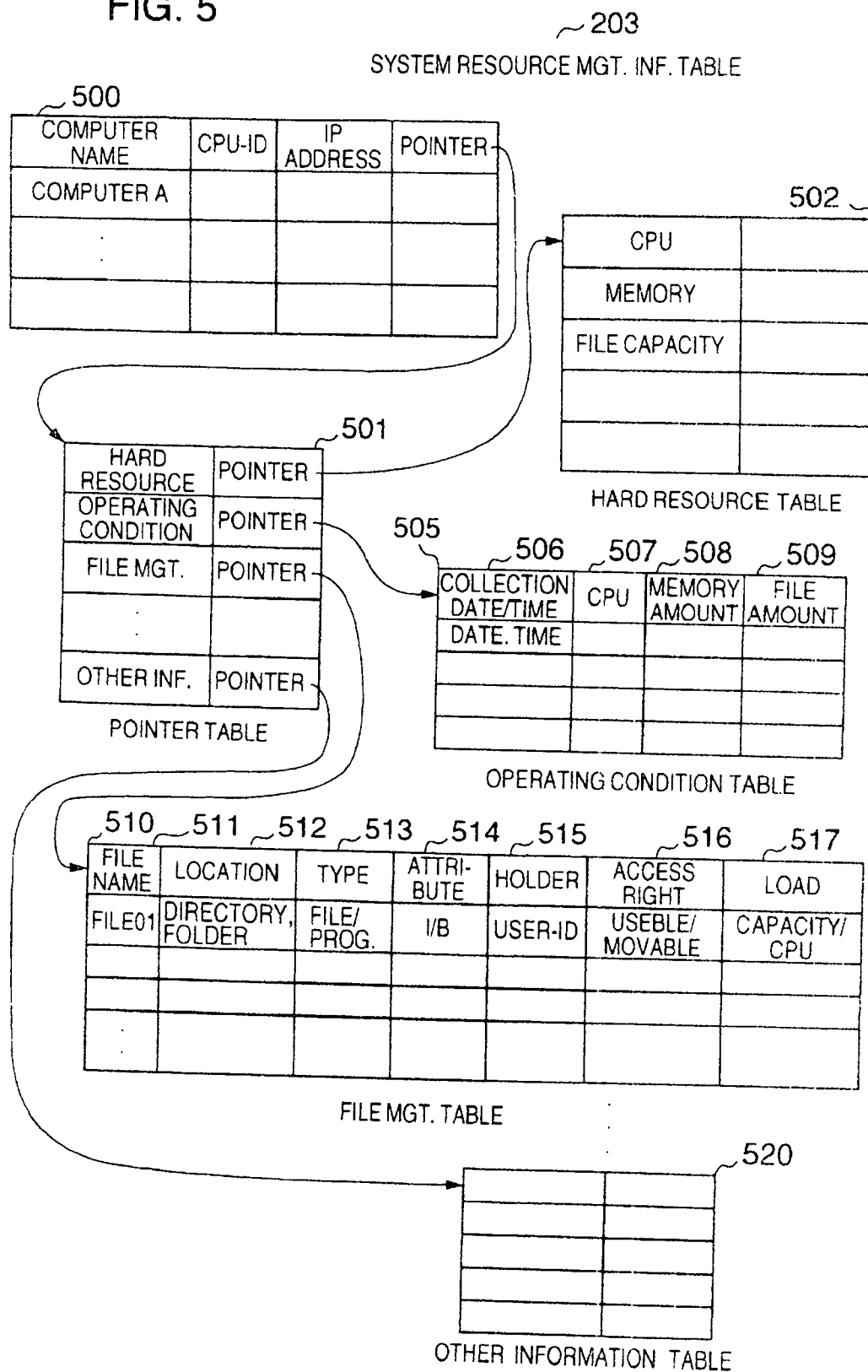
392

FIG. 4



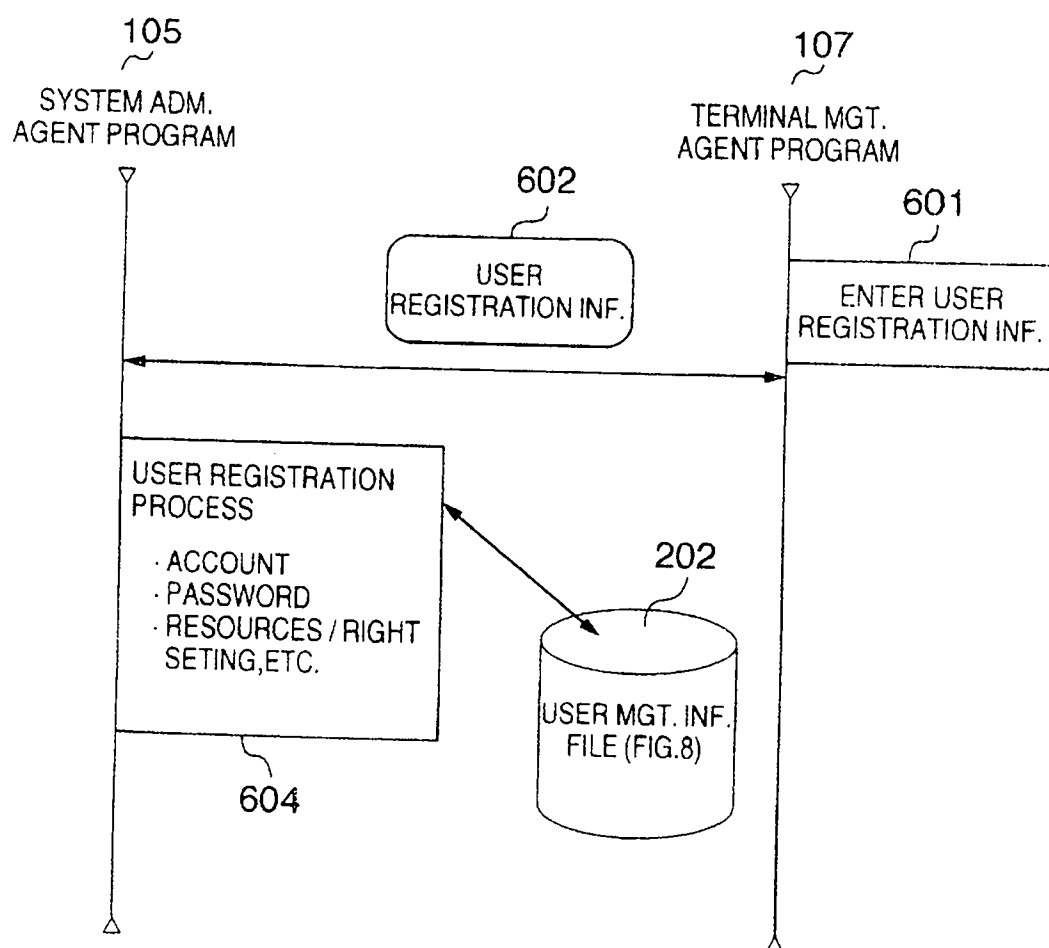
393

FIG. 5



394

FIG. 6



395

FIG. 7

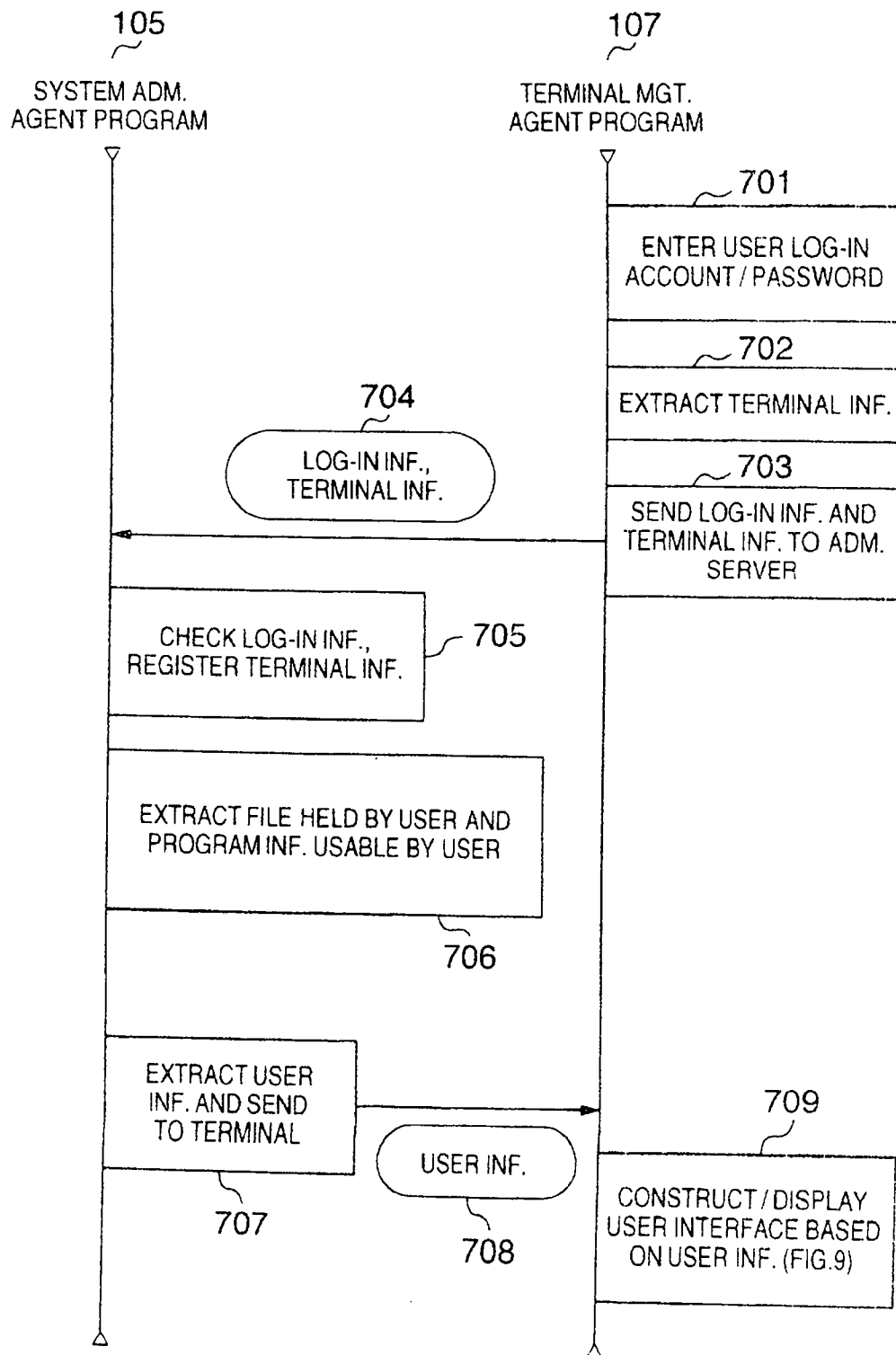
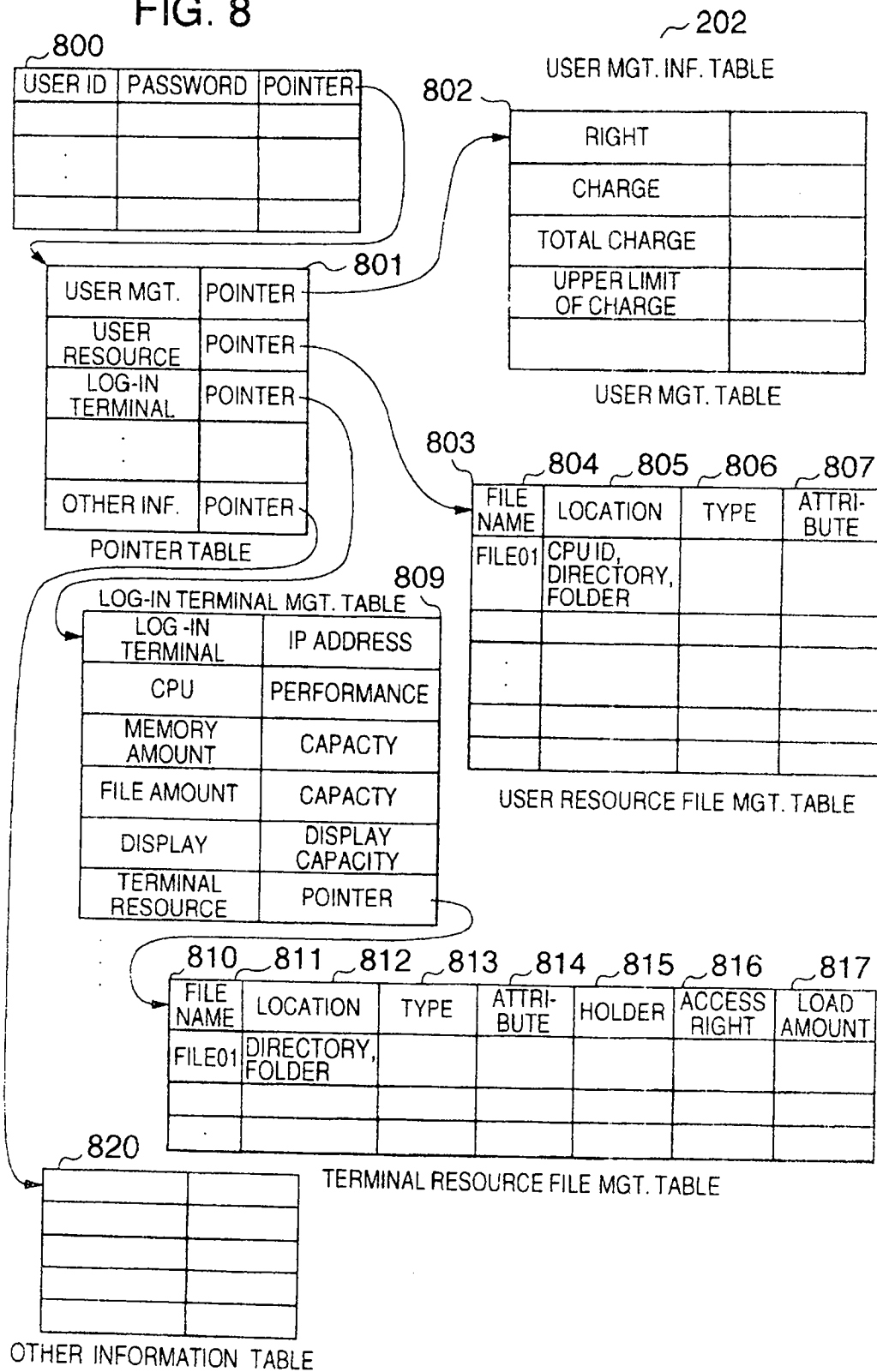
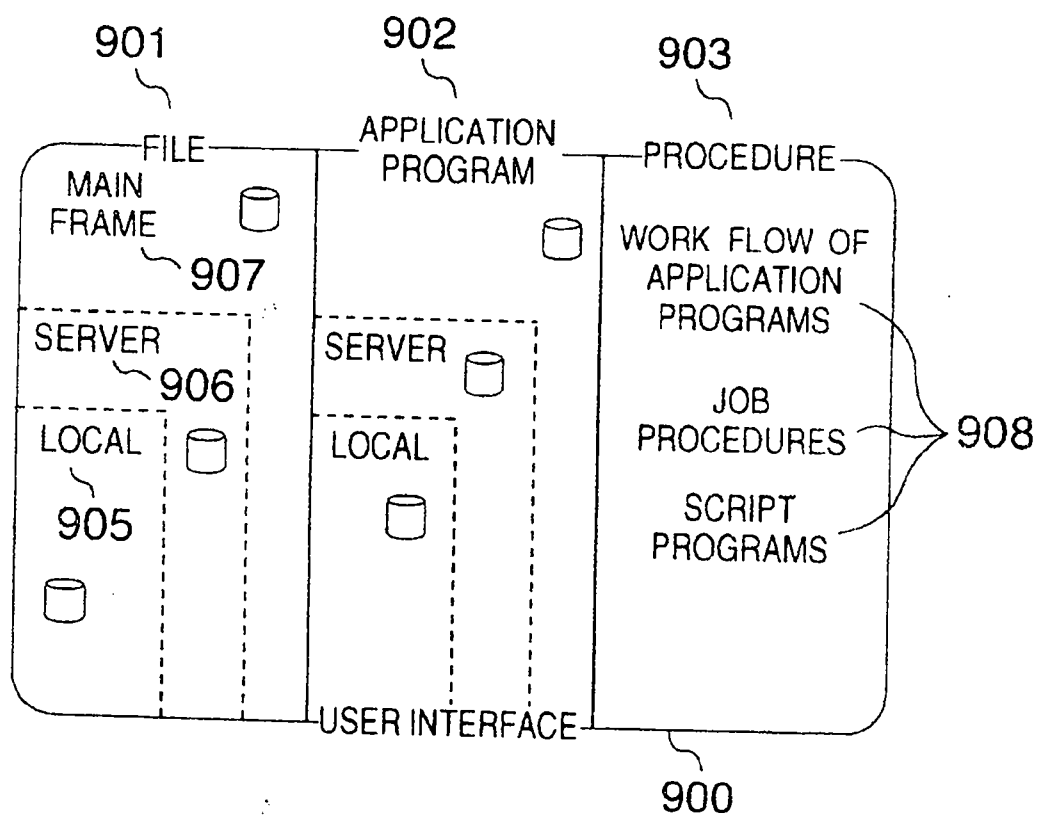


FIG. 8



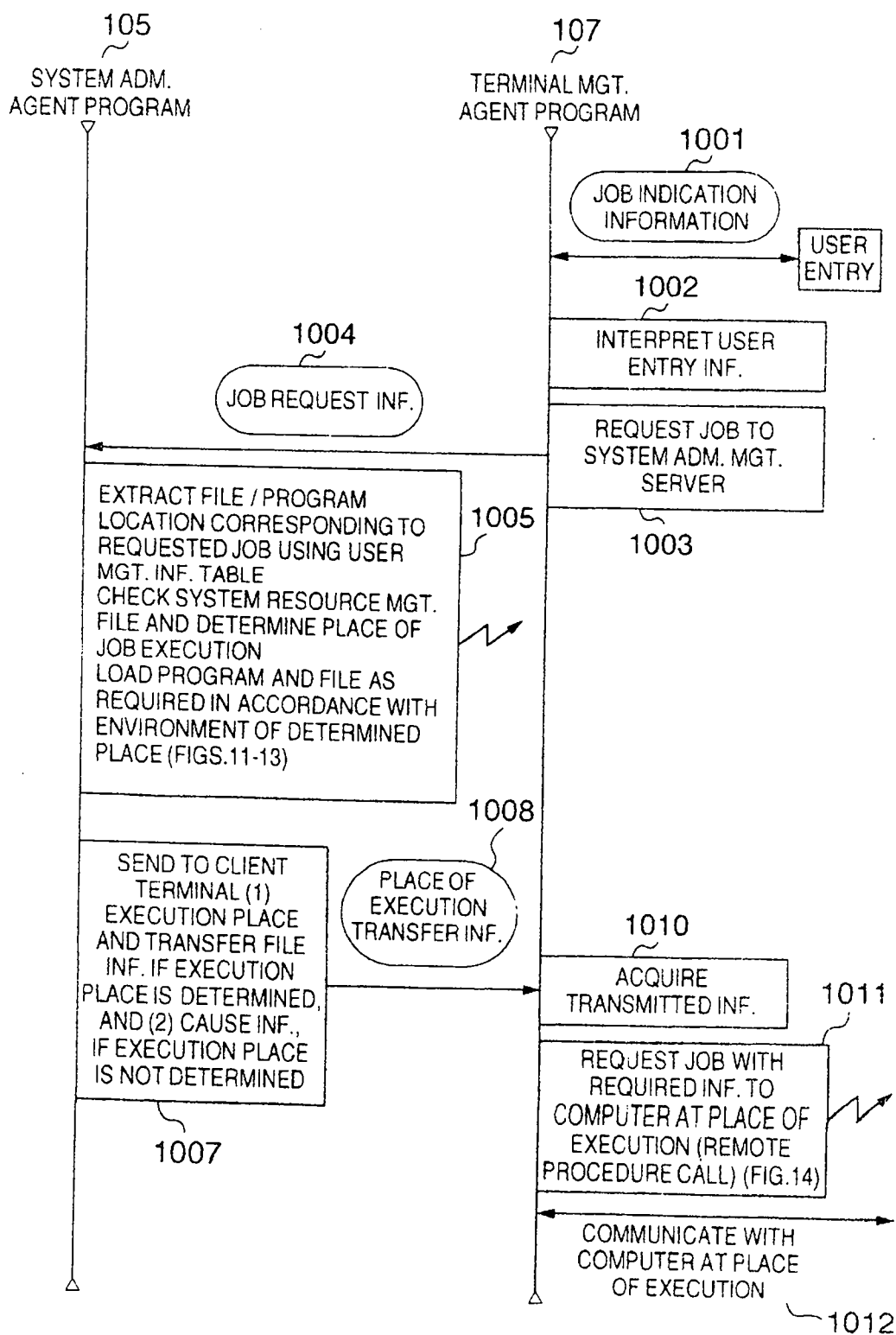
397

FIG. 9



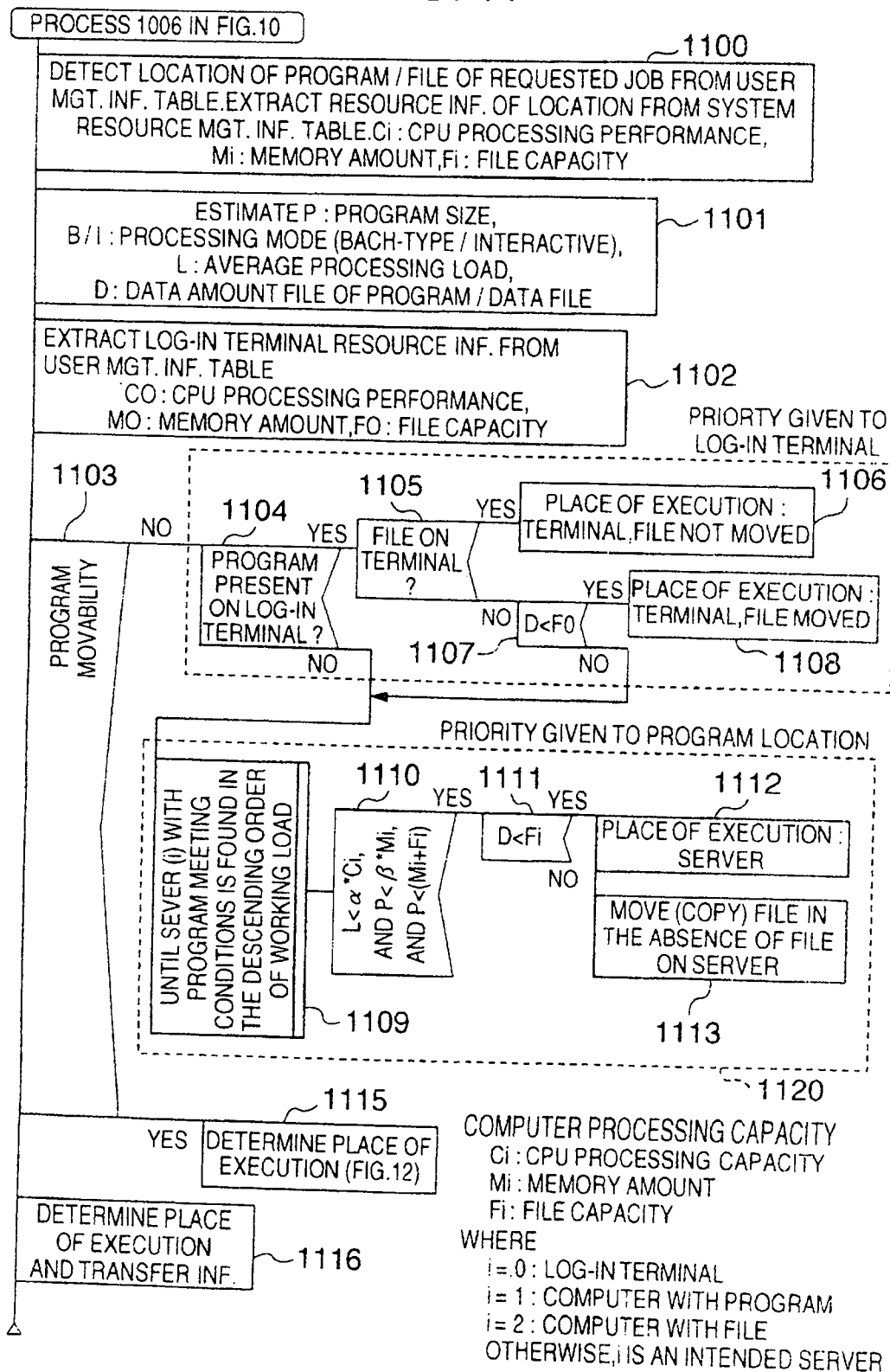
398

FIG. 10



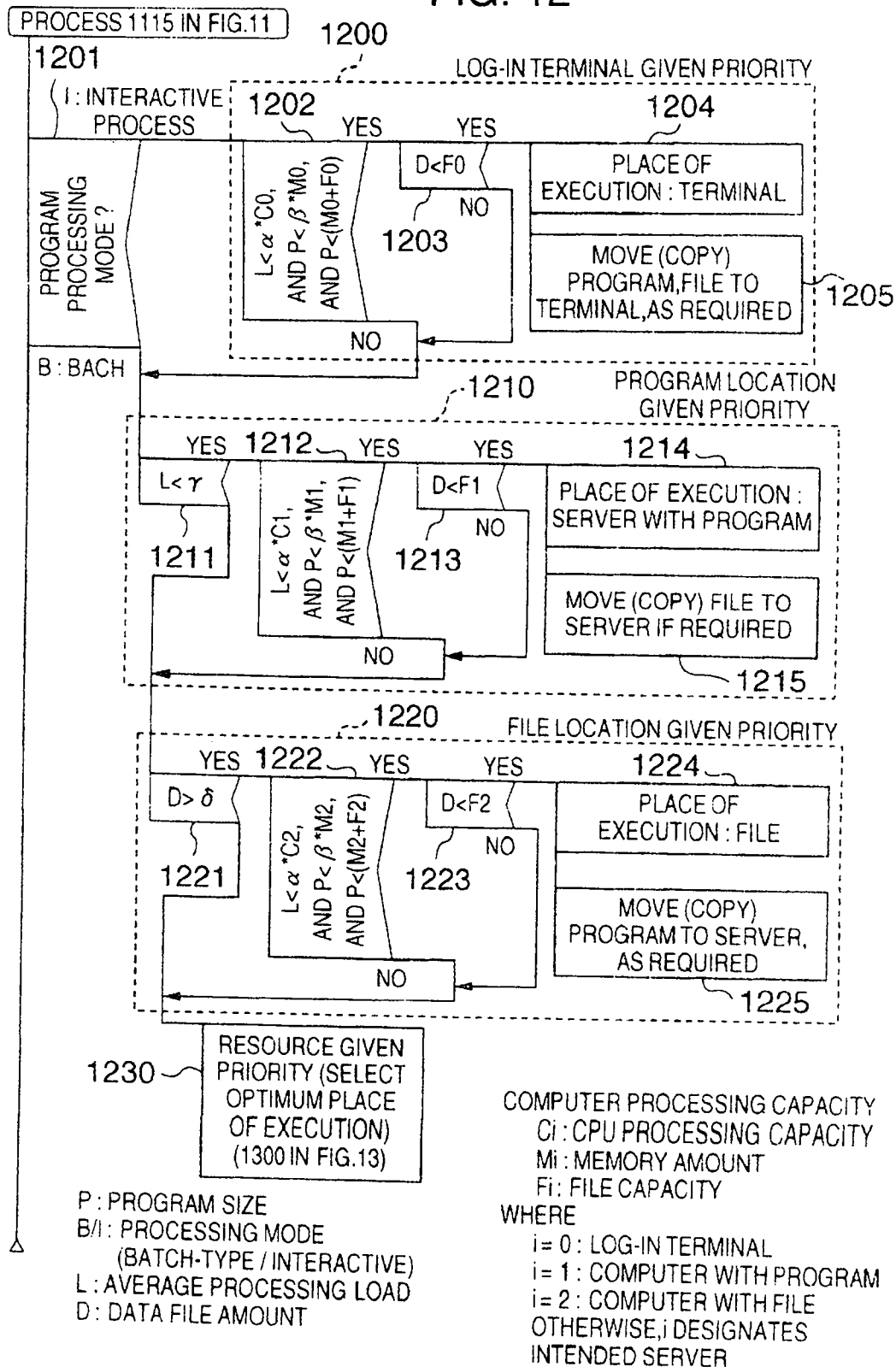
399

FIG. 11



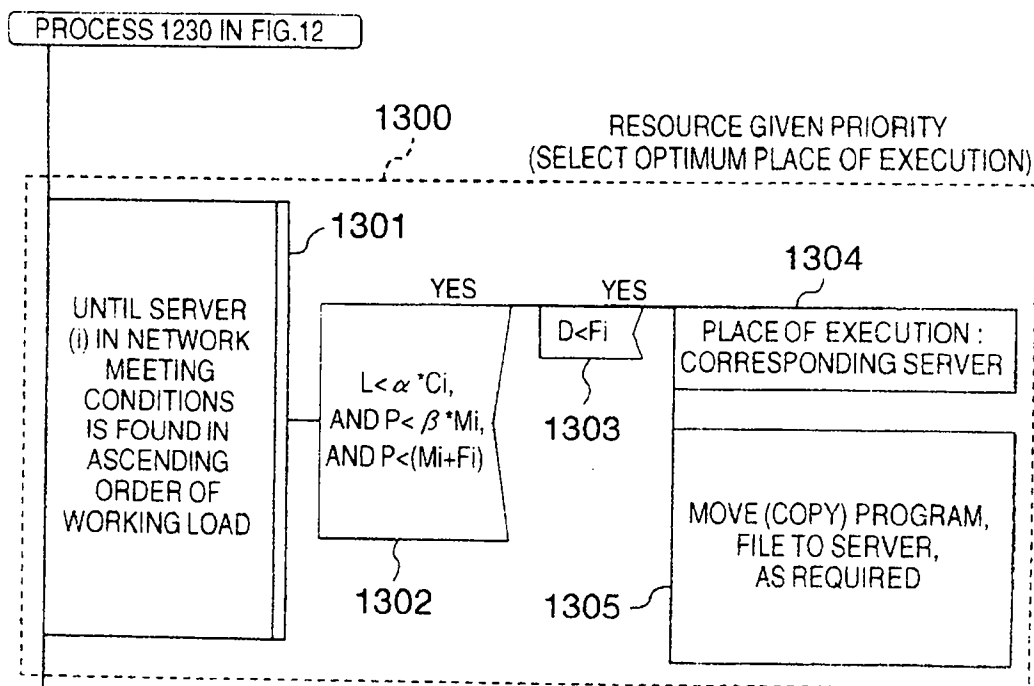
400

FIG. 12



401

FIG. 13



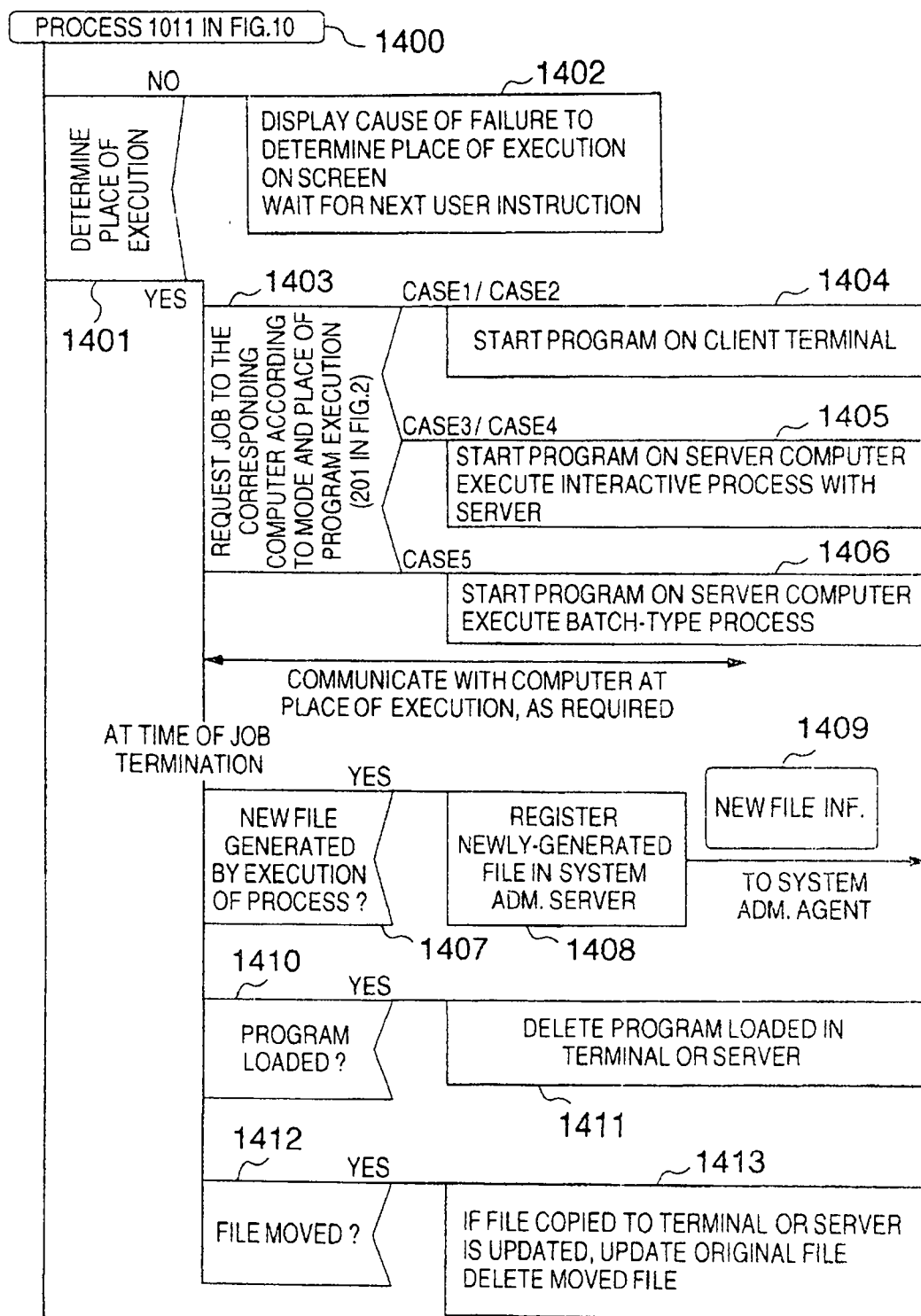
COMPUTER PROCESSING CAPACITY
 C_i : CPU PROCESSING CAPACITY
 M_i : MEMORY AMOUNT
 F_i : FILE CAPACITY

WHERE

$i = 0$: LOG-IN TERMINAL
 $i = 1$: COMPUTER WITH PROGRAM
 $i = 2$: COMPUTER WITH FILE
 OTHERWISE, i INDICATES AN INTENDED SERVER

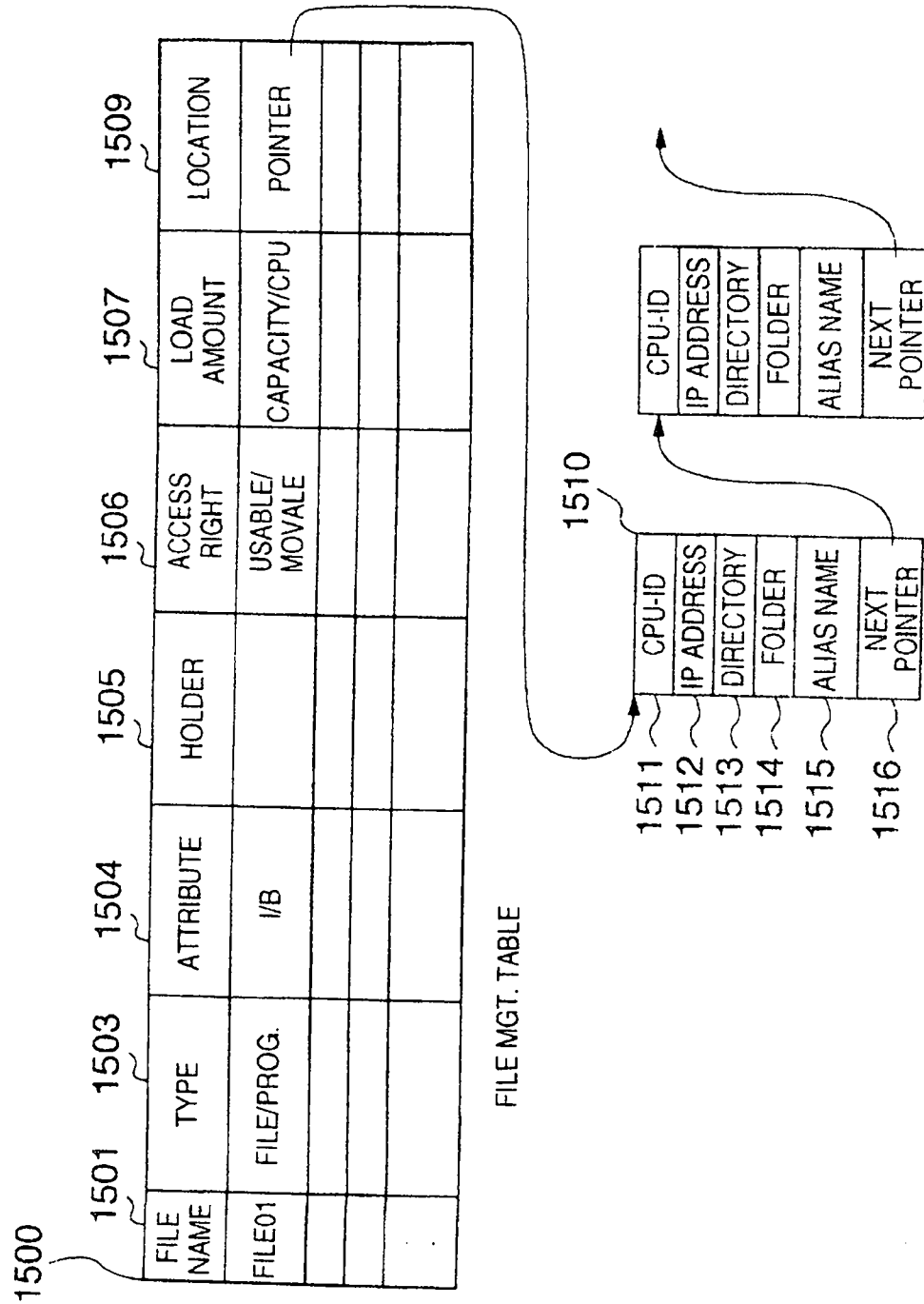
402

FIG. 14



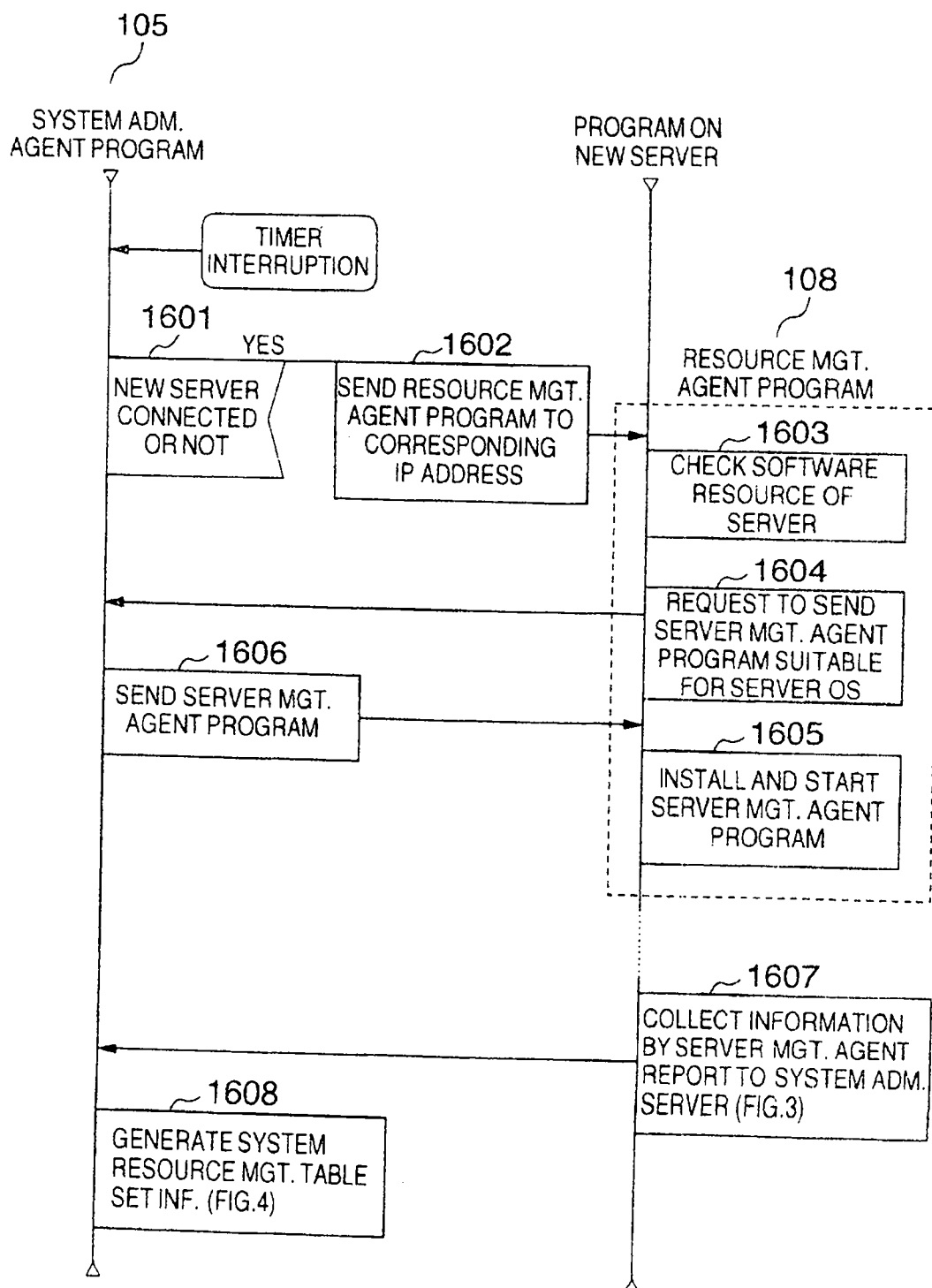
403

FIG. 15



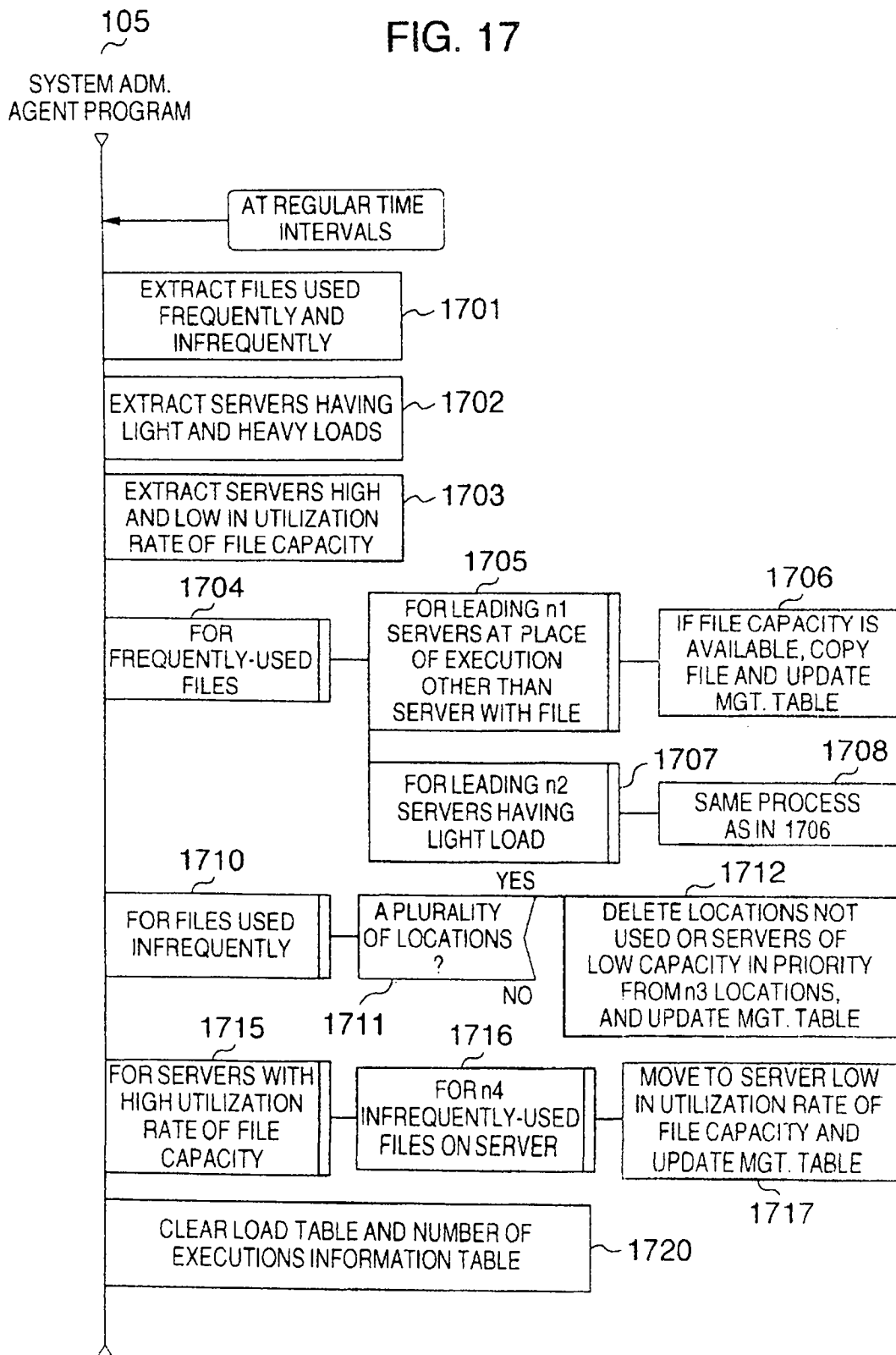
464

FIG. 16

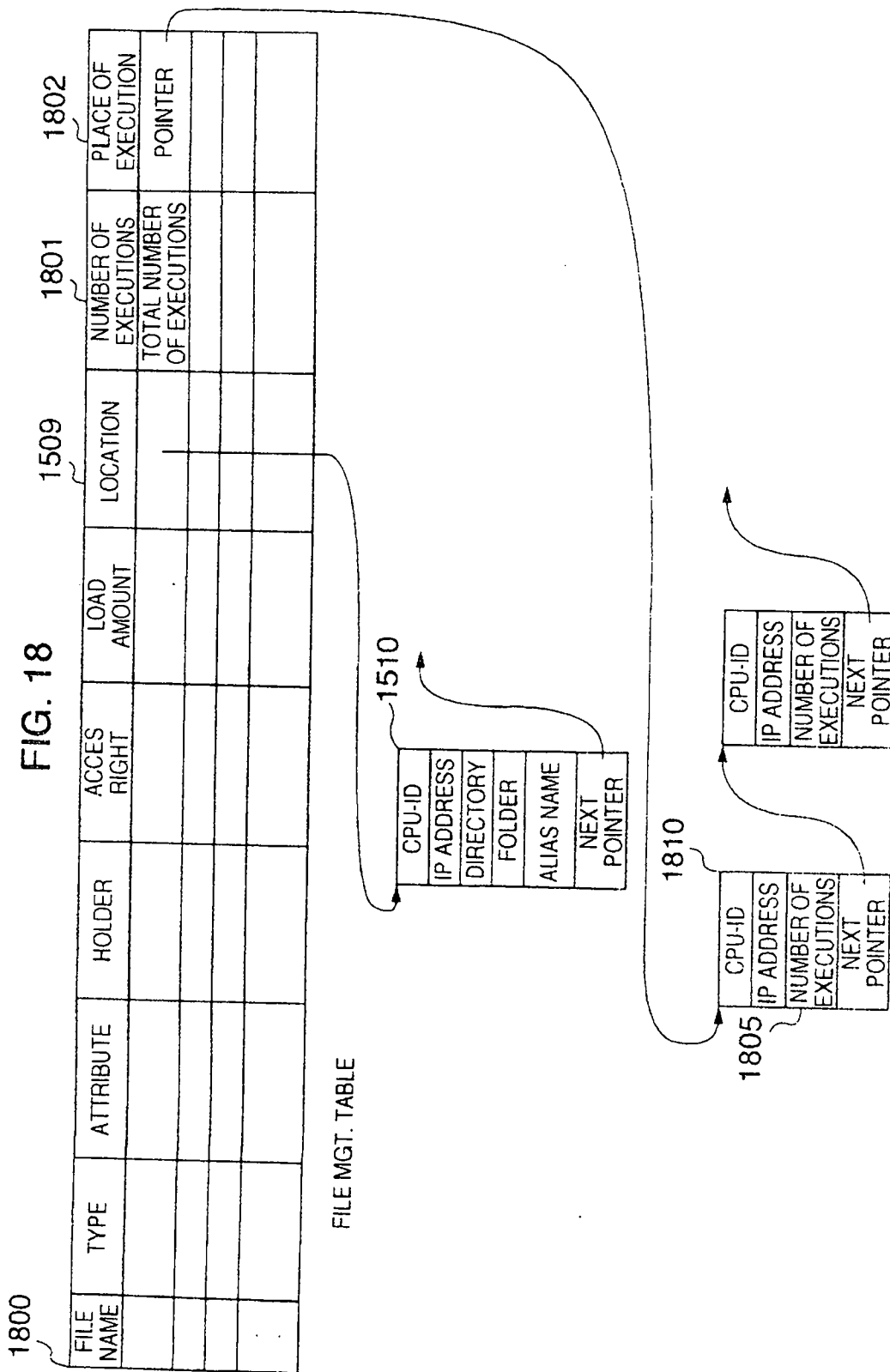


405

FIG. 17



406



United States Patent [19]
Miller et al.

US005640569A

[11] Patent Number: 5,640,569
[45] Date of Patent: Jun. 17, 1997

- [54] DIVERSE GOODS ARBITRATION SYSTEM AND METHOD FOR ALLOCATING RESOURCES IN A DISTRIBUTED COMPUTER SYSTEM
- [75] Inventors: Mark S. Miller; E. Dean Tribble, both of Los Altos; Norman Hardy, Portola Valley; Christopher T. Hibbert, Mountain View, all of Calif.
- [73] Assignee: Sun Microsystems, Inc., Mountain View, Calif.
- [21] Appl. No.: 431,021
- [22] Filed: Apr. 28, 1995
- [51] Int. Cl.⁶ G06F 13/366; G06F 15/76
- [52] U.S. Cl. 395/729; 395/237
- [58] Field of Search 395/729, 208, 395/237, 650

[56] References Cited

U.S. PATENT DOCUMENTS

4,789,928 12/1988 Fujisaki 395/237
5,243,515 9/1993 Lee 395/237
5,394,324 2/1995 Clearwater 395/237
5,487,168 1/1996 Geiner et al 395/208

OTHER PUBLICATIONS

"An Algorithm for Optimal Service Provisioning Using Resource" Steven Low et al., IEEE 1994.
Post et al., "Application of Auctions as a Pricing Mechanism for the Interchange of Electric Power", IEEE Transactions on PWR System, vol. 10, No. 3, Aug. 1995.
Lee, "Electronic Brokerage and Electronic Auction: The Impact of IT on Market Structure", IEEE 1996.
M. Anderson, R.D. Pose, C.S. Wallace, "A Password-Capability System", The Computer Journal, vol. 29, No. 1, 1986, pp. 1-8.
Mark S. Miller, K. Eric Drexler, "Comparative Ecology: A Computational Perspective", The In The Ecology of Computation, Bernardo Huberman (ed.) Elsevier Science Publishers/North-Holland, 1988, pp. 1-26.

Mark S. Miller, K. Eric Drexler, "Markets and Computation: Agoric Open Systems", In The Ecology of Computation, Bernardo Huberman (ed.) Elsevier Science Publishers/North-Holland, 1988, pp. 1-44.
K. Eric Drexler, Mark S. Miller, "Incentive Engineering: for Computational Resource Management", In The Ecology of Computation, Bernardo Huberman (ed.) Elsevier Science Publishers/North-Holland, 1988, pp. 1-36.

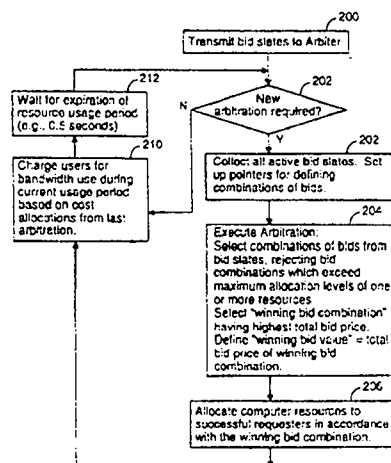
(List continued on next page.)

Primary Examiner—Gopal C. Ray
Assistant Examiner—Xuong M. Chung-Trans
Attorney, Agent, or Firm—Gary S. Williams; Flehr Hohbach Test Albritton & Herbert LLP

[57] ABSTRACT

A diverse goods arbitration system and method allocates computer resources among bidding requesters. Bid slates are transmitted to an arbiter by users (requesters) requesting use of specified portions of the available computer resources. Each bid slate may contain a plurality of bids, each bid representing a requested set of resources and a bid price. The arbiter selects combinations of bids from the bid slates, where each bid combination consists of no more than one bid from each of the received bid slates. The arbiter rejects all bid combinations whose constituent bids exceed an established maximum allocation level for any computer resource. It then selects as a winning bid combination the bid combination having the highest total bid price. Computer resources are then allocated for a next time period based on the winning bid. Costs are allocating to each successful requester in accordance with a predefined opportunity cost function. In particular, for each successful requester, the arbitration process is repeated while excluding that successful requester's bid slate from the set of bid slates considered, resulting in the selection of a second winning bid that excludes the successful requester. The successful requester is then assessed a cost corresponding to the difference between the winning bid's total bid prices, excluding the price in the successful requester's granted bid, and the total bid prices associated with the second winning bid.

8 Claims, 4 Drawing Sheets



OTHER PUBLICATIONS

C.S. Wallace, R.D. Pose. "Charging in a Secure Environment". Bremen, FRG., May 1990, pp. 24-1 through 24-11.

Michael P. Wellman, "A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems", AI Access Foundation and Morgan Kaufmann Publishers, Aug. 1993, pp. 1-23.

22
408

DIVERSE GOODS ARBITRATION SYSTEM AND METHOD FOR ALLOCATING RESOURCES IN A DISTRIBUTED COMPUTER SYSTEM

The present invention relates generally to multiprocessor computer systems, distributed computer systems and networks of interconnected computer systems, and particularly to a system and method for allocating resources among computer system users competing for those resources using a technique herein called diverse goods arbitration.

BACKGROUND OF THE INVENTION

The cheaper a resource becomes, the more important it becomes to have automatic management of that resource in a principled fashion. This is because the increased capacity makes it possible to apply that resource to lower-value uses. When a resource is expensive, all uses, in order to be worth the amount they consume, must exceed some minimum value to their users, so all uses of an expensive resource have high and comparable values. Also, because individual uses of an expensive resource have high value, it is both easy and worthwhile to manage the resource manually. When the price per unit of that resource drops by multiple orders of magnitude, uses that have a low value per resource unit become feasible, but these low-value uses can crowd out the high-value uses if there is no way to express the value differences—and manual management of the resource becomes prohibitively difficult, as well as expensive relative to the value of the resource.

For example, the transition to fiber data communication makes possible bandwidth-intensive applications like video. It might seem, since fiber has so much more bandwidth than previous communications media, that existing network allocation policies (like those used in Ethernet) can simply be scaled. However, in the case of Ethernet, the amount of bandwidth that an application is allocated is proportional to how much it asks for and how often it asks. In such a regime, network users trying to transmit, for example, electronic mail may find network performance significantly reduced by the presence of video users, even though both the data users and video users would agree that the value per bit of electronic mail is far higher than that of video data: a megabyte of capacity would carry an entire day's correspondence for a large company, but amount to only a single frame of high-resolution video, which the video user could drop without noticing. Expanding the overall capacity of the network will not address the problem, because the higher capacity of the network invites still more video use. Non-argoric protocols provide no way to express this value difference. Thus, economic abstractions, such as price and value, meaningful to the system and automatically taken into account, are needed in order to manage abundant computing resources.

The primary existing resource allocation methods for allocating a computer system resource are: (1) first come first served, (2) priority based allocation methods, (3) pro rata allocation methods (allocate in proportion to amount requested), and (4) simple "single good" auction methods. In "first come first served" systems, bandwidth is assigned in the order that requests are made, until there is insufficient bandwidth left to satisfy further requests, at which point the requests that cannot be satisfied are simply denied. In priority based allocation methods, higher priority requests displace lower priority requests when there is insufficient bandwidth to satisfy all requests.

For systems having a single allocatable system resource (such as bandwidth on a transmission medium) and many requesters, such as the system shown in FIG. 1, the prior art literature discloses a technique that will for the purposes of this document be called "single good auctions." In a single good auction, each requester specifies the amount of the good being requested, and a bid price that the requester is willing to pay. The arbiter (sometimes called the auctioneer) selects the set of bids that maximize the payment received, and each successful requester whose request is granted pays the amount that the requester bid.

There are several different ways in which such single good auctions can be conducted. Conventional auction techniques include the English auction, also known as the first-price, open-bid auction, in which the highest price bid is the price paid and every bidder knows the bids made by others. Another auction mechanism is the Dutch auction, in which a seller offers lower and lower prices until a buyer or set of buyers claims the item at the last offered price. In a double auction, sellers offer lower and lower prices while buyers offer higher and higher prices until they meet. In a first-price sealed-bid auction, fixed bids are submitted and the highest is accepted. In a second-price sealed-bid auction, the highest bid is accepted, but the highest bidder pays the amount bid by the second-highest bidder.

While the prior art has provided many well proven systems for allocating a single resource through various auction and arbitration systems, those systems are not appropriate for the simultaneous arbitration of multiple goods, or "baskets of goods," where the goods being allocated are relatively inexpensive and in such great supply that the requests of numerous requesters can be satisfied simultaneously. For instance, one might think that each of the goods could be allocated separately through a separate arbitration, using one of the known arbitration mechanisms. However, the inadequacy of separate arbitrations is demonstrated as follows. A user who requests 8 Mb/sec of bandwidth on three sequentially connected links needs to be allocated exactly the same amount of bandwidth on all three links, or none at all. Allocating this user 8 Mb/sec of bandwidth on two of the links and none on the third link is of zero value to the user, because without a bandwidth allocation on the third link he will receive no signals whatsoever.

Thus, it is an object of the present invention to provide a diverse goods arbitration mechanism that preserves the integrity of each diverse goods bid and either allocates all the goods in a bid, in the quantities requested, or denies the bid.

Another object of the present invention is to provide an efficient and fair system and method for allocating a plurality of computer resources in an integrated arbitration, and for allocating costs to the successful requesters in accordance with the opportunity costs associated with each successful request for computer resources.

SUMMARY OF THE INVENTION

In summary, the present invention is a diverse goods arbitration system and method for allocating computer resources among bidding requesters. The present invention's arbitration methodology uses a variant of the second-price sealed-bid auction. This arbitration mechanism is efficient in that each good is sold or allocated to the right buyer and the right price. Note that the goal of the arbitration mechanism is not to maximum monetary return to the arbiter, but rather to allocate resources to maximum declared value. Successful requesters are charged based on opportunity costs, which

27
278
278
409

best reflects the economic cost of allocating system resources to each successful requester.

In accordance with the present invention, bid slates are transmitted to an arbiter by users (requesters) requesting use of specified portions of the available computer resources. Each bid slate may contain a plurality of bids, each bid representing a requested set of resources and a bid price.

The arbiter stores data representing a maximum allocation level for each computer resource and stores additional data representing bid slates it receives from the requesters. The arbiter selects combinations of bids from the bid slates, where each bid combination consists of no more than one bid from each of the received bid slates. The arbiter rejects all bid combinations whose constituent bids exceed the maximum allocation level of any computer resource. From among the non-rejected bid combinations, it then selects as a winning bid combination the bid combination having the highest total bid price. Computer resources are then allocated for a next time period based on the winning bid.

Costs are allocated to each successful requester in accordance with a predefined opportunity cost function. In particular, for each successful requester, the arbitration process is repeated while excluding that successful requester's bid slate from the set of bid slates used, resulting in the selection of a second winning bid that excludes the successful requester. The successful requester is then assessed a cost corresponding to the difference between the winning bid's total bid prices, excluding the price bid by the successful requester, and the total bid prices associated with the second winning bid.

In a preferred embodiment, the computer resources being allocated by the arbiter are the transmission bandwidth of a multiplicity of network links interconnecting the users with one or more video servers as well as an electronic mail or file server.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

FIG. 1 is a block diagram of an Ethernet style prior art system.

FIG. 2 is a block diagram of a switch based communication network utilizing a diverse goods agoric arbiter.

FIG. 3 is a block diagram of an arbiter in accordance with a preferred embodiment of the present invention.

FIG. 4 schematically represents a simple distributed computer system having a switch based communication network utilizing a diverse goods agoric arbiter.

FIG. 5 depicts a bid slate used in a preferred embodiment.

FIG. 6 is a flow chart of the arbitration method employed by a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 2, a typical system in which the present invention is used is a distributed computer system 100 having multiple signal sources 102, numerous subscribers (sometimes called users or requesters) 104, and a switch based communication network 106 that interconnects them. In the preferred embodiment, the signal sources 102 include video servers and electronic mail servers.

Bandwidth allocation requests made by the subscribers 104 constitute competing requests for multiple goods

because the available bandwidth on each network link represents a distinct and separate good. The requests made will typically include bandwidth requests for communication paths that are at least partially overlapping.

The term "agoric" is used in this document to mean "utilizing market mechanisms, such as auctions, for resource allocation in computer systems."

The diverse goods agoric arbiter 110 receives bandwidth allocation requests from subscribers 104. The arbiter 110 then selects a combination of bids, called the winning bid combination, that maximizes the total bid price for all the winning bids, while not violating any established maximum resource allocation levels. The arbiter then transmits resource grant messages that allocate system resources in accordance with the winning bid combination.

The resource allocation process performed by the arbiter is implemented as a continuing arbitration process. The resource allocations resulting from an arbitration are maintained if no new bids have been received, no continuing bids have been withdrawn, and there has been no change in the network configuration. Whenever at least one new bid is received, any continuing bids is withdrawn, or there has been a change in the network configuration, the arbitration procedure is re-executed. More particularly, each arbitration cycle results in a resource allocation for a fixed period of time, such as 0.5 seconds. Every time period, the arbiter 110 re-executes the arbitration procedure if there are any new requests, any previously pending requests have been withdrawn, or there has been a network configuration change. Otherwise the previous resource allocation is maintained.

Referring to FIG. 3, the arbiter 110 in the preferred embodiment includes a central processing unit (CPU) 120, a network interface 122 for receiving bid slates from network subscribers, and local memory 124. The local memory 124 stores the arbitration procedure 130 that implements the methodology of the present invention, an array of data 132 representing the bid slates 134 received from subscribers and another array of data 140 representing the allocation status of each allocatable computer resource. Array 140 includes an allocation record 142 for each allocatable resource indicating the maximum allocation level for the corresponding resource and the amount of the resource allocated by the currently selected bid combination.

In addition, local memory 124 is also used to store a highest total bid price 144, representing the total bid value of the best bid found so far, a current bid position vector 146, representing a currently selected bid combination, and a best bid combination vector 148, representing the best bid combination found so far during the arbitration process.

Bids for allocation of resources are submitted by subscribers to the arbiter in the form of "bid slates." Each bid slate 134 identifies the requester, and contains a sequence of one or more bids. Each bid in the bid slate includes a specification of a requested combination of goods, and an associated bid price. For instance, a bid might include quantity QA for good A, quantity QB for good B and so on. By allowing each subscriber to submit a bid slate having multiple bids, the subscriber can indicate that while a first resource allocation combination would be most valuable to the subscriber, other resource allocation combinations would be of somewhat less value to the subscriber. For instance, the subscriber might be willing to pay \$5 per hour for full bandwidth video channel, but only \$2 per hour for a compressed bandwidth video channel. Each such "channel" will typically be represented by a combination of bandwidth allocations on a number of network links.

The simplified distributed computer system 160 shown in FIG. 4 will be used to demonstrate operation of the present invention. This distributed computer system has a switch based communication network and a diverse goods agoric arbiter 110. In this system, there are two signal sources, A-S and C-S, and three subscribers who can submit resource allocation requests, B-R, D-R and G-R. The simplified network has just two switches, E-SW and F-SW, and six communication links, Link1 through Link6. The signal sources A-S and C-S are video servers with multiple levels of service. Programs and connections to the video servers can be provided at multiple bandwidths and quality levels. In addition, for purposes of this explanation of the invention we will assume that signal source A-S also functions as an electronic mail server, as well as a video server.

For simplicity, it will be assumed that the maximum allowed allocation for each individual link (Link1-Link6) is 10 "units," as 10 megabytes per second of bandwidth.

In the simplified system of FIG. 4, a somewhat modified Bid Slate format is used, as shown in FIG. 5. In particular, each bid slate 162 used in the system of FIG. 4 includes a representation of the requester's identification, a signal path indicator, followed by two or more "bandwidth, price" pairs. Rather than representing a basket of goods in each bid record, the basket of goods for each bid is represented by a combination of the path indicator and the bandwidth parameter. For instance, in the following bid slate subscriber B is requesting a communication connection from node A to node B at three bandwidth levels: 0 Mb/sec, 8 Mb/sec and 5 Mb/sec.

BID SLATE

Requester: B	
Path: A to B	
Bid 1: 0	0
Bid 2: 8	5
Bid 3: 5	2

Thus, the second bid in this exemplary bid slate represents a request for 8 Mb/sec of bandwidth on links 1, 3 and 4, and thus represents a request for specific quantities of three goods, at a bid price of 5 monetary units per unit of time (e.g., 5 dollars per hour).

For purposes of explanation, in the system of FIG. 4 we will assume that all goods requests are stated in units of megabytes of bandwidth per second, and that all bid prices are rates of payment amount per time period, and in particular dollars per hour. In other embodiments of the invention, the computer resources to be allocated might include memory resources as well as link bandwidth resources and perhaps other resources as well (e.g., computation time on a compute server).

Also, for purpose of this explanation, requests will be considered to be continuing requests until withdrawn. Bids are retained and used in all resource allocation arbitrations until withdrawn. Arbitrations are held only when new requests are received, when any previously received request is withdrawn, or if the set of available computer resources changes.

The arbitration procedure used in the preferred embodiment will be explained with reference to the flow chart in FIG. 6 and a pseudocode representation of the procedure in Table 1. The pseudocode used in Table 1 is, essentially, a computer language using universal computer language conventions. While the pseudocode employed here has been invented solely for the purposes of this description, it is designed to be easily understandable by any computer programmer skilled in the art.

The first step (200) of the arbitration process is for the arbiter to receive bid slates that have been transmitted by requesters, to determine if any previously submitted bids have been withdrawn, and to construct a data structure 132 in local memory representing all the pending bid slates.

If no new bid slates have been received since the last arbitration, and no continuing bids have been withdrawn (202), the resource allocations from the prior arbitration are maintained for an additional resource allocation period and the users of those allocated resources are changed for one more usage period in accordance with the charge rates determined during the last arbitration (210, 212).

If at least one new bid slate has been received, or at least one bid slate has been withdrawn (202), or the network configuration has changed, a special bid combination pointer vector 146 (see FIG. 3) is defined. In one preferred embodiment, this pointer vector 146 is implemented as an "N stage counter."

The purpose of the bid combination pointer vector is to point, in sequence, to every possible combination of bids. Each bid combination in the preferred embodiment includes one bid from each pending bid slate. This simplified bid combination point vector structure is made possible because each bid slate 162 includes a zero bid for no resources. It is not necessary for the requesters to include a zero bid in their transmitted bid slates; a zero bid is added by the arbiter if one was not included by the requester.

Each stage of the N stage counter references one bid in the corresponding bid slate. For instance, if there are three pending bid slates, the first two of which include three bids each and the last of which includes two bids, then the N stage counter will have three stages. The first two stages of the 3 stage counter will increment from 0 to 1 to 2 before cycling back to zero, while the last stage will increment from 0 to 1 before cycling back to zero. Thus, in essence, the N stage counter is implemented as a set of N counters connected in tandem: as each counter is incremented, it causes the next higher order counter to increment only when it cycles back to a value of zero. Each stage of the counter can have a different maximum value, depending on the number of bids in the corresponding bid slate.

The arbitration process (204) for selecting a winning bid combination is performed as follows. Starting with an N stage counter value of zero, every possible bid combination, or at least a substantial subset of the possible bid combinations, is evaluated in two respects. In particular, for each bid combination to be evaluated, a total resource allocation request value is generated for all of the allocatable goods, with the computed values being stored in their respective positions in the goods allocation array 140. If the total resource allocation request value for any good exceeds the maximum allowed allocation of that good, the corresponding bid combination is rejected. If the bid combination is not rejected, the total bid price for the bid combination is computed and compared with the highest bid total (144) of any bid combination evaluated so far. If the current bid combination's total bid price exceeds the previous highest bid total, the current bid combination is stored as the best bid combination (so far) (148) and the highest bid total (144) is updated to represent the current bid combination's total bid price.

This process is repeated for either all possible values of the N stage counter, or a selected subset of the possible values. In many circumstances, an evaluation of the bid slates and the extents to which the resources they are requesting overlap will allow the "search tree" of bid combinations to be substantially pruned, thereby reducing

the amount of computation performed by the arbiter. For instance, in many circumstances, once certain bid combinations have been determined to violate certain maximum resource allocation levels, other bid combinations with even higher resource allocation requests do not need to be explicitly evaluated since it is known in advance that they will also violate the same maximum resource allocation levels. In another example, the bids within the bid slates may be re-arranged so that bid combinations can be evaluated in decreasing order of total bid price, such that once a bid combination is found that does not violate any resource allocation limits, that bid combination is used as the winning bid combination without further analysis of any further bid combinations.

Once the bid combinations have been evaluated, the best bid combination vector 148 represents the winning bid combination.

For each non-zero bid in the winning bid combination, the arbiter next computes an opportunity cost which will be charged to the corresponding requester for each usage period. Each non-zero bid in the winning bid combination is considered to be a "granted request." For purposes of explaining the computation of each winning bid's associated opportunity cost, we will assume that there are M such granted requests.

For each of the M granted requests (where $M \leq N$) the arbitration procedure is re-executed. Each time the arbitration procedure is re-executed, the bid slate of one of the winning requesters is excluded from the arbitration. The winning requester is charged a price proportional to the difference between the value of the winning bid combination for the reduced requester arbitration and the value of the winning bid combination for the full arbitration, excluding the price in the successful requester's granted bid:

$$\begin{aligned} \text{Opportunity Cost (Successful requester i)} \\ &= (\text{Reduced Arbitration Total Bid Price}) \\ &\quad - (\text{Full Arbitration Total Bid Price} \\ &\quad \quad - \text{Successful requester i's Winning Bid Price}) \\ \text{Cost per Resource Usage Period} \\ &= \beta \times \text{Opportunity Cost (Successful requester i)} \end{aligned}$$

The opportunity cost for each successful requester will range between zero, and the full bid price in the successful requester's winning bid. This can be seen from the following simple analysis. First, if removal of the successful requester's slate from the arbitration does not change any of the other winning bids, the successful requester's request has not caused anyone else's bid to be rejected or reduced. Thus, the opportunity cost for the successful requester in this circumstance is zero.

Second, the reduced arbitration cannot produce a new total bid price greater than the total bid price of the full arbitration. If the reduced arbitration produces a total bid price equal to the total bid price of the full arbitration, which is the worst case scenario, the opportunity cost associated with the successful requester is his/her full bid price, since his/her bid has resulted in reduced allocations to others have a value equal to his/her full bid price.

Except in certain special cases, such as where all maximum bids are included in the winning bid combination (and thus the opportunity cost is zero for all successful requesters), the analysis of the opportunity cost must be performed separately for each successful requester.

Once the winning bid combination, and the associated opportunity costs, have been determined by the arbiter, the arbiter transmits resource allocation messages for allocating

system resources in accordance with the winning bid combination and for charging the successful requesters in accordance with the determined opportunity costs. In particular, for each non-zero bid in the winning bid combination, the arbiter transmits a resource grant message to the winning requester and debits a resource account for the winning requester for each resource usage period by an amount corresponding to the computed opportunity cost. As shown above, the cost per resource usage period is determined by multiplying the computed opportunity cost by a time factor β . For instance, if opportunity costs are determined on a usage per hour basis, and the resource usage period is 0.5 seconds, then the computed opportunity costs for each successful requester are each divided by 7200 to determine the allocated cost per resource usage period.

DIVERSE GOODS ARBITRATION EXAMPLE

Referring to FIGS. 4 and 5, for purpose of demonstrating the operation of the invention, we will assume that three bid slates have been submitted:

Bid Slate 1	Bid Slate 2	Bid Slate 3
Requester: B	Requester: D	Requester: G
Path A-B	Path C-D	Path A-G
Bid 0: 0 0	Bid 0: 0 0	Bid 0: 0 0
Bid 1: 8 7	Bid 1: 6 4	Bid 1: 1 3
Bid 2: 3 2	Bid 2: 3 2	

Bid slates 1 and 2 are requests for video connections, each at two distinct bandwidth/quality levels, while Bid slate 3 is for a file transfer or electronic mail connection. Note that in this arbitration, the allocations of six goods are being arbitrated simultaneously: the bandwidth allocations of Link1, Link2, . . . Link6. In this particular example, two of the link bandwidths (for Link1 and Link3) are the subject of competing bids. In more realistic commercial settings, many links will typically be the subject of competing bids during heavy usage periods.

The preferred embodiment of the arbitration procedure evaluates all possible bid combinations, as follows:

3-Stage Counter Value	Violates Max?	Total Bid Price
000	No	0
001	No	3
010	No	4
011	No	7
100	No	7
101	No	10
110	Yes	—
111	Yes	—
120	Yes	—
121	Yes	—
200	No	2
201	No	3
210	No	6
211	No	9
220	No	4
221	No	7

Thus, the winning bid combination is bid 1 from bid slate 1 and bid 1 from bid slate 3, having a total bid price of 10.

The opportunity costs for successful requesters B and G are determined as follows. To determine the opportunity cost associated with requester B's successful request, the arbitration is re-executed without bid slate 1. This results in a second winning bid of bid 1 from bid slate 2 and bid 1 from bid slate 3, with a total bid price of 7. As a result the opportunity cost to successful requester B is:

$$\text{Opportunity Cost}(B) = 7 - (10 - 7) = 4$$

The result, 4, makes intuitive sense in that the second arbitration had a total bid price 3 units less than in the full arbitration, meaning that 3 units of B's success request price of 7 did not represent a opportunity cost.

The same procedure is next applied to determine the opportunity cost associated with requester G's successful request, but re-executing the arbitration without bid slate 3. This results in a second winning bid of bid 1 from bid slate

illustrative of the invention and is not to be construed as limiting the invention. Various modifications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

TABLE 1

Pseudocode Representation of Diverse Goods Agoric Arbitration procedure

```

Repeat Forever
{
  If no new bids have been received, no continuing bids have been withdrawn,
  and there has been no change in the network configuration
  {
    Charge successful requesters from prior arbitration for one usage
    period
    Wait for expiration of resource usage period
  }
  Else
  {
    Initialize N stage counter to zero
    Initialize best bid vector and best bid value to zero
    Do Until N Stage Counter reaches final value
    {
      If corresponding bid violates none of the bandwidth constraints
      {
        Determine value of corresponding bid
        If determined value > best bid value
          Best bid value = determined value
          Best bid vector = bid vector corresponding to current
          value of N Stage Counter
      }
    }
    /* End Do Loop */
    Initialize Opportunity Costs for all requesters (i=1 to N) to zero
    M = number of requests that were Granted
    WR(1:M) = vector identifying the M requesters whose bids were
    granted
    WB(1:M) = vector identifying the M winning bids
    If Best Bid Vector = Maximum value of N Stage Counter
    Return /* Opportunity Cost for all bidders is zero when all
    bidders were granted their highest bandwidth request */
    Do for i = 1 to M
    {
      ReRun Arbitration(i,j) /* Arbitration excludes requester WR(i) */
      /* j = best bid value for rerun arbitration */
      /* If vector of granted requests is the same as the Best Bid
      Vector, except for W(i), OpportunityCost(i) = 0 */
      OpportunityCost(i) = j - (Best Bid Value - WB(i).BidPrice)
    }
    /* End Do Loop */
    Transmit resource allocation messages for allocating system resources
    in accordance with the winning bid combination.
    Charge each of the successful requesters a price proportional to the
    computed OpportunityCost for that bidder for each resource
    allocation period that the resource allocation is maintained.
    Wait for expiration of resource usage period.
  }
}

```

1, with a total bid price of 7. As a result the opportunity cost to successful requester G is:

$$\text{Opportunity Cost}(G) = 3 - (10 - 7) = 0$$

The result, 0, makes intuitive sense because the second arbitration produces the exact same result as the full arbitration, except for requester G's bid. This means that there was no opportunity cost associated with G's successful request.

While the present invention has been described with reference to a few specific embodiments, the description is

What is claimed is:

1. A method of allocating a plurality of computer resources among a plurality of users, comprising the steps of:

(A) transmitting to an arbiter bid slates from ones of said users (requesters) requesting use of specified portions of said computer resources, at least a plurality of said bid slates including a plurality of bids, each bid in each bid slate representing a requested set of resources, composing a specified subset of said plurality of computer resources, and a bid price;

- (B) at said arbiter:
- (B1) storing data representing a maximum allocation level for each of said plurality of computer resources;
 - (B2) receiving said bid slates;
 - (B3) selecting for analysis combinations of bids from said bid slates, where each bid combination consists of no more than one bid from each of said received bid slates, rejecting all those of said bid combinations whose constituent bids include requests for one or more of said computers resources in excess of the corresponding maximum allocation levels for those computer resources, and selecting as a winning bid combination one non-rejected bid combination that includes bids which place highest value on use of said plurality of computer resources;
 - (B4) allocating to each successful requester having a non-zero bid included in said winning bid combination a cost in accordance with a predefined opportunity cost function; and
- (C) allocating said computer resources among said successful requesters in accordance with said winning bid.
2. The method of claim 1 including, at said arbiter:
- (B5) defining a winning bid value as the total bid price of the bids in said winning bid combination; and
 - (B6) for each successful requester:
 - (B6A) performing said selecting step an additional time while excluding said each successful requester's bid slate from the set of bid slates used to form a test set of bid combinations to select a second winning bid that excludes said each successful requester; and
 - (B6B) allocated to said each successful requester a cost corresponding to the difference between said winning bid value and the total bid prices associated with said second winning bid.
3. The method of claim 1, wherein said plurality of computer resources includes transmission bandwidth on a multiplicity of network links interconnecting said users.
4. The method of claim 1, wherein
- said plurality of computer resources includes transmission bandwidth on a multiplicity of network links interconnecting said users and at least one video server;
 - said transmitted bid slates include a first and second bids in a first bid slate where said requested set of resources are first and second bandwidth allocation requests on a plurality of said network links, both sufficient for real time transmission of a motion picture video program, and having a first and second bid price, respectively, a third bid in another bid slate where said requested set of resources is a third bandwidth allocation request on a plurality of said network links that is insufficient for real time transmission of a motion picture video program, having a third bid price;
 - said first, second and third bandwidth allocation requests overlap on at least one network link;
 - said first bandwidth allocation is greater than said second bandwidth allocation and said second bandwidth allocation is greater than said third bandwidth allocation;
 - said bid prices each represent a value for use of the requested bandwidth allocation for a period of time of less than 5 seconds duration, and wherein said third bid price is greater than said second bid price; and
 - said winning bid combination selected at said arbiter includes said third bid in said winning combination whenever said first bid is included in said winning combination.

5. An arbiter for allocating a plurality of computer resources among a plurality of users, comprising:
- a receiver for receiving bid slates from ones of said users (requesters) requesting use of specified portions of said computer resources, at least a plurality of said bid slates including a plurality of bids, each bid in each bid slate representing a requested set of resources, composing a specified subset of said plurality of computer resources, and a bid price;
 - arbitration apparatus, including:
 - memory for storing data representing a maximum allocation level for each of said plurality of computer resources and for storing said received bid slates;
 - bid selection means for selecting combinations of bids from said bid slates, where each bid combination consists of no more than one bid from each of said received bid slates, rejecting all those of said bid combinations whose constituent bids include requests for one or more of said computers resources in excess of the corresponding maximum allocation levels for those computer resources, and selecting as a winning bid combination one non-rejected bid combination that includes bids which place highest value on use of said plurality of computer resources; and
 - cost allocation means for allocating to each successful requester having a non-zero bid included in said winning bid combination a cost in accordance with a predefined opportunity cost function; and
 - a transmitter for transmitting one or messages that allocate said computer resources among said successful requesters in accordance with said winning bid.
6. The arbiter of claim 5,
- said bid selection means defining a winning bid value as the total bid price of the bids in said winning bid combination; and
 - said cost allocation means including means for allocating a cost to each successful requester by:
 - performing said selecting step an additional time while excluding said each successful requester's bid slate from the set of bid slates used to form a test set of bid combinations to select a second winning bid that excludes said each successful requester; and
 - allocated to said each successful requester a cost corresponding to the difference between said winning bid value and the total bid prices associated with said second winning bid.
7. The arbiter of claim 5, wherein said plurality of computer resources includes transmission bandwidth on a multiplicity of network links interconnecting said users.
8. The arbiter of claim 5, wherein
- said plurality of computer resources includes transmission bandwidth on a multiplicity of network links interconnecting said users and at least one video server;
 - said transmitted bid slates include a first and second bids in a first bid slate where said requested set of resources are first and second bandwidth allocation requests on a plurality of said network links, both sufficient for real time transmission of a motion picture video program, and having a first and second bid price, respectively, a third bid in another bid slate where said requested set of resources is a third bandwidth allocation request on

415 ~~0000~~
~~0001~~

5,640,569

13

a plurality of said network links that is insufficient for
real time transmission of a motion picture video
program, having a third bid price;
said first, second and third bandwidth allocation requests
overlap on at least one network link;
said first bandwidth allocation is greater than said second
bandwidth allocation and said second bandwidth allo-
cation is greater than said third bandwidth allocation;

14

said bid prices each represent a value for use of the
requested bandwidth allocation for a period of time of
less than 5 seconds duration, and wherein said third bid
price is greater than said second bid price; and
said winning bid combination selected by said bid selec-
tion means includes said third bid in said winning
combination whenever said first bid is included in said
winning combination.

* * * * *

416

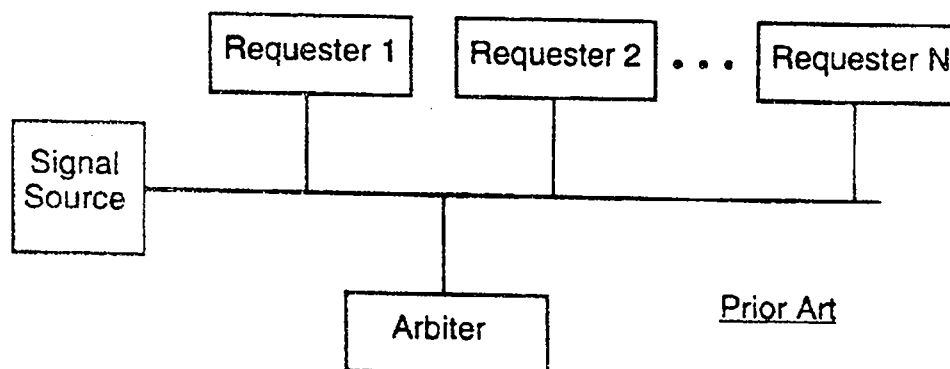


FIGURE 1

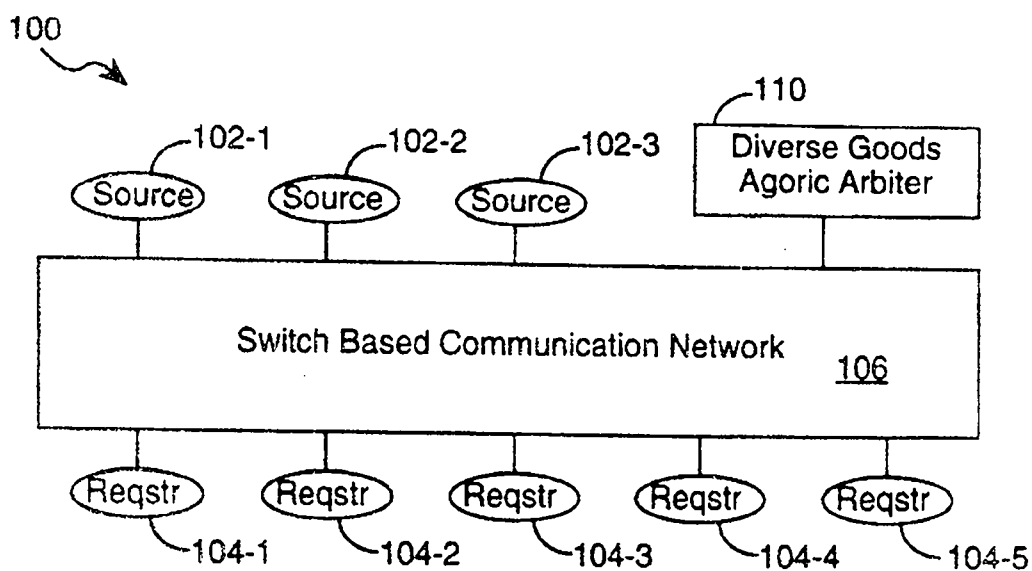


FIGURE 2

417

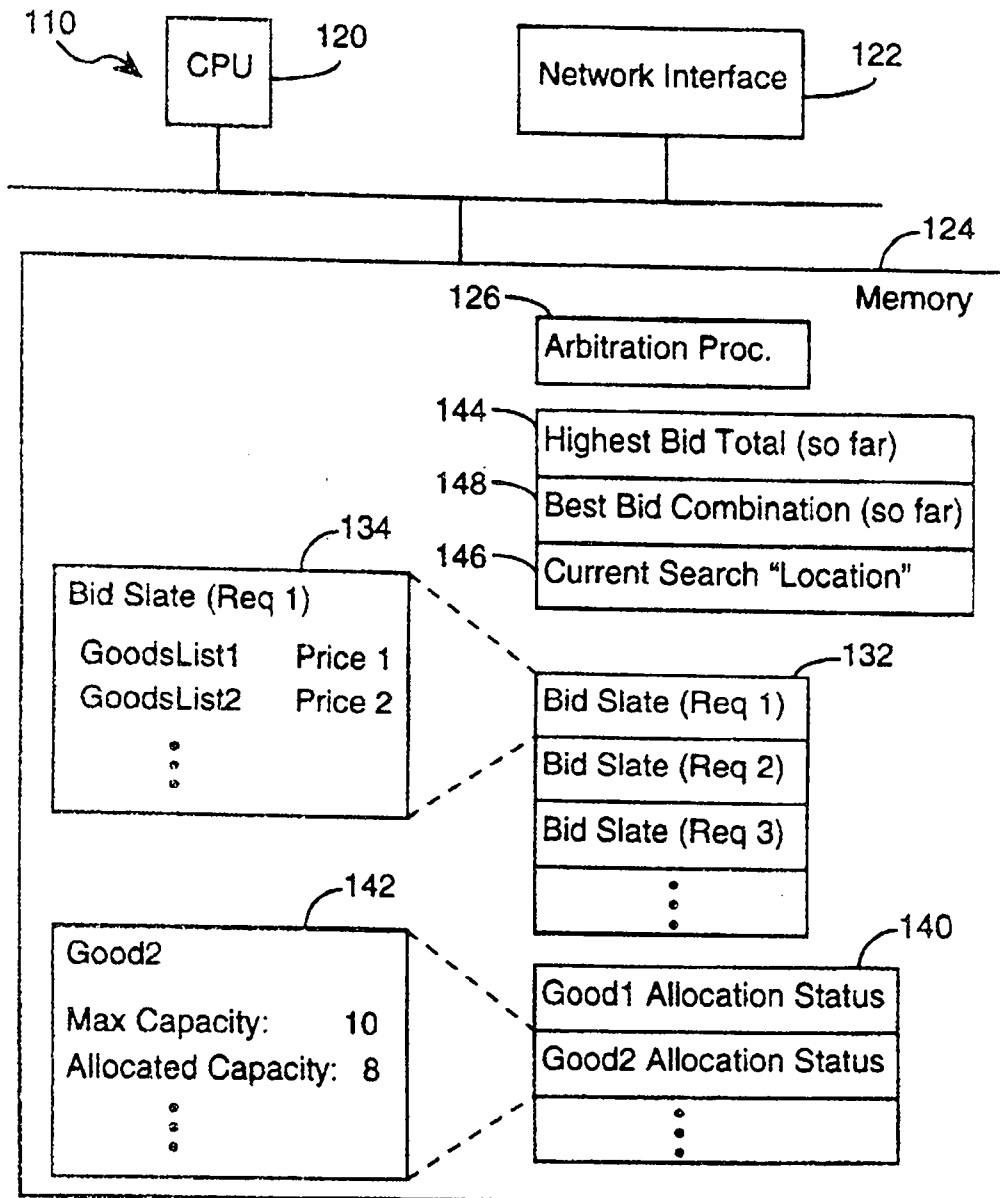


FIGURE 3

418 ~~418~~

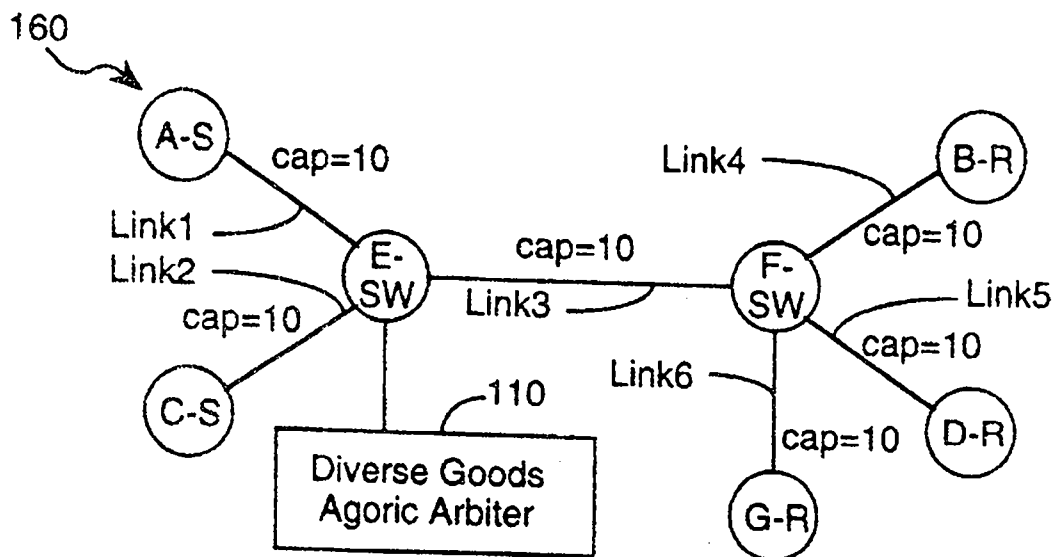


FIGURE 4

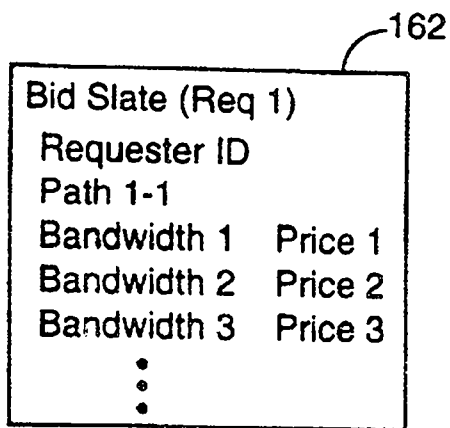


FIGURE 5

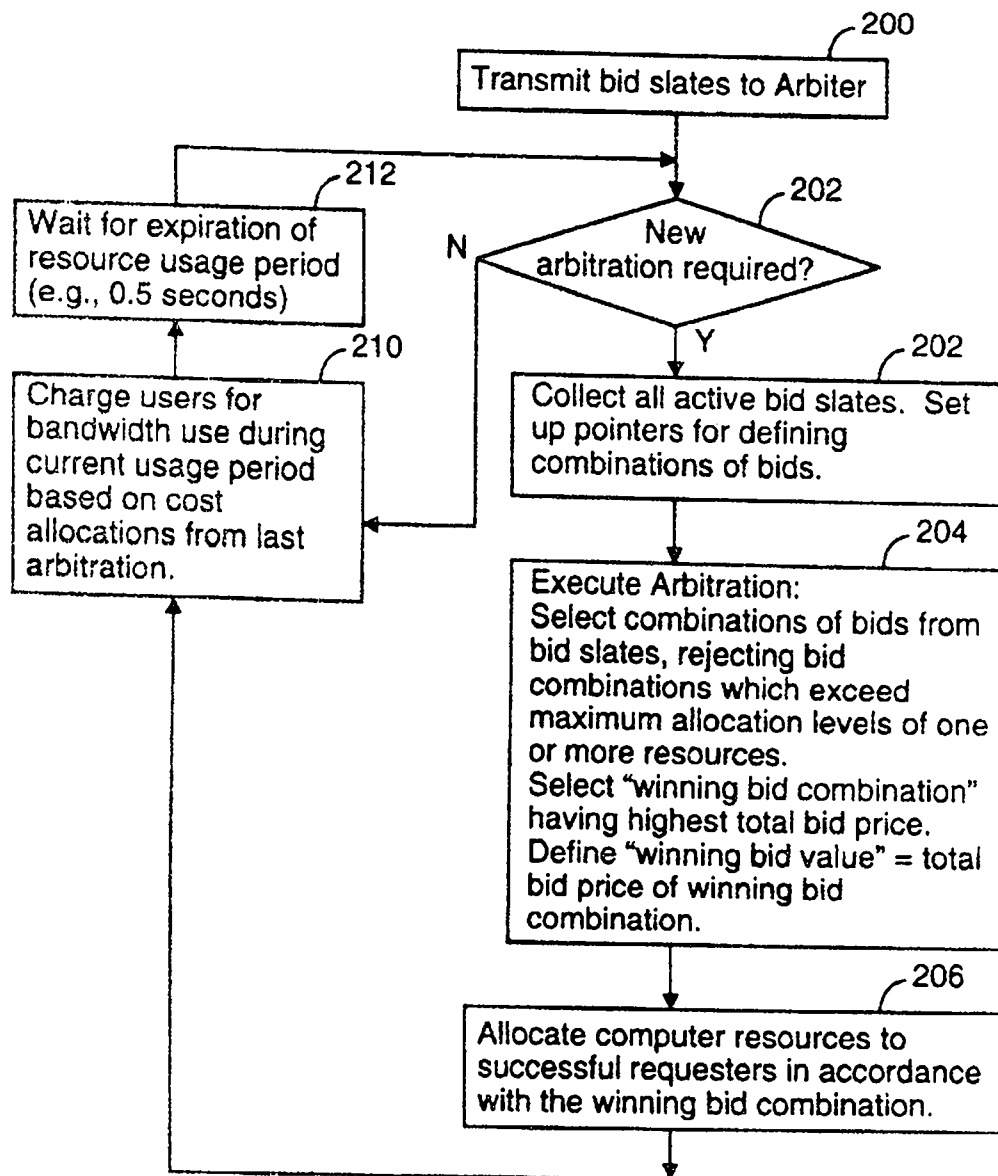


FIGURE 6



US006247026B1

(12) United States Patent
Waldo et al.**(10) Patent No.: US 6,247,026 B1**
(45) Date of Patent: *Jun. 12, 2001**(54) METHOD, APPARATUS, AND PRODUCT FOR LEASING OF DELEGATION CERTIFICATES IN A DISTRIBUTED SYSTEM**

4,713,806 12/1987 Oberlander et al. 370-358

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0 300 516	1/1989 (EP)	
0 351 536 A2	1/1990 (EP)	G06F 9/46
0 351 536 A3	1/1990 (EP)	G06F 9/46
0 384 339	8/1990 (EP)	
0 472 874	3/1992 (EP)	
0 474 340	3/1992 (EP)	
0 497 022 A1	8/1992 (EP)	
0 555 997	8/1993 (EP)	

(List continued on next page.)

OTHER PUBLICATIONS

Chung, Kin-Man and Yuen, Herbert. "A 'Tiny' Pascal Compiler: Part 1: The P-Code Interpreter," BYTE Publications, Inc., pp. 58-155, Sep. 1978.

Chun, Kin-Man and Yuen, Herbert. "A 'Tiny' Pascal Compiler: Part 2: The P-Compiler," BYTE Publications, Inc., pp. 34-52, Oct. 1978.

Thompson, Ken. "Regular Expression Search Algorithm," Programming Techniques, Communications of the ACM, vol. 11, No. 6, pp. 419-422, Jun., 1968.

(List continued on next page.)

(75) Inventors: James H. Waldo, Dracut; Ann M. Wollrath, Groton; Robert Scheffler, Somerville, all of MA (US)**(73) Assignee:** Sun Microsystems, Inc., Palo Alto, CA (US)**(*) Notice:** This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

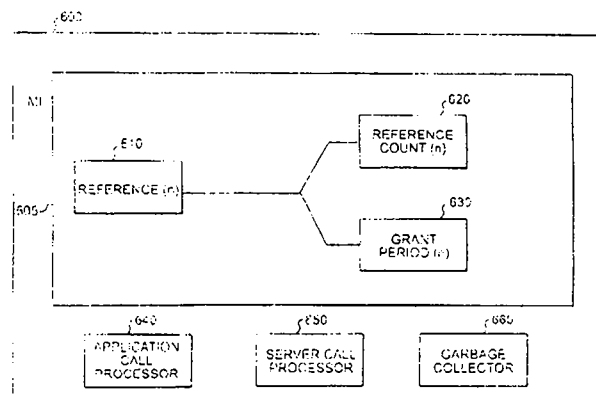
This patent is subject to a terminal disclaimer.

(21) Appl. No.: 09/044,838**(22) Filed: Mar. 20, 1998****Related U.S. Application Data****(63)** Continuation-in-part of application No. 08/729,421, filed on Oct. 11, 1996, now Pat. No. 5,832,529.**(60)** Provisional application No. 60/076,048, filed on Feb. 26, 1998.**(51) Int. Cl.⁷** G06F 17/30**(52) U.S. Cl.** 707/206; 707/9; 707/10; 707/201; 707/205; 709/104; 709/203**(58) Field of Search** 707/10, 104, 9, 707/201, 205, 206; 709/104, 203**(56) References Cited****U.S. PATENT DOCUMENTS**

4,491,946 1/1985 Kryskow, Jr. et al. 370/453

Primary Examiner—Jack Choules**Assistant Examiner**—Cheryl Lewis**(74) Attorney, Agent, or Firm**—Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.**(57) ABSTRACT**

A method, apparatus, and product for leasing a delegation certificate in a distributed processing system is provided. Consistent with this method, apparatus, and product, a delegatee requests from a delegator access to a resource for a period of time that the delegator has authority to access. Responsive to this request, the delegator determines an appropriate lease period during which time the delegatee has authority to access to the resource.

46 Claims, 13 Drawing Sheets

U.S. PATENT DOCUMENTS

4,809,160	2 1989	Mahon et al.	713/200
4,823,122	4 1989	Mann et al.	340/825.28
4,939,638	7 1990	Stephenson et al.	710,244
4,986,773	9 1990	Saito et al.	717/3
5,058,036	2 1992	Ellis et al.	
5,107,486	4 1992	Seymour	709/224
5,187,787	2 1993	Skeen et al.	395/600
5,218,699	6 1993	Brandle et al.	709/328
5,257,369	10 1993	Skeen et al.	395/650
5,293,614	3 1994	Ferguson et al.	395/600
5,297,283	3 1994	Kelly, Jr. et al.	709/104
5,311,591	5 1994	Fischer	713/156
5,339,435	8 1994	Lubkin et al.	717/11
5,386,568	1 1995	Wold et al.	717/10
5,390,328	2 1995	Frey et al.	709/315
5,423,642	6 1995	Jafili et al.	709/328
5,440,744	8 1995	Jacobson et al.	709/203
5,448,740	9 1995	Kiri et al.	717/8
5,452,450	9 1995	Drury et al.	707/3
5,455,982	10 1995	Gjovaag	717/1
5,471,629	11 1995	Risch	707/201
5,475,792	12 1995	Stanford et al.	704/233
5,475,817	12 1995	Waldo et al.	709/316
5,481,721	1 1996	Serlet et al.	709/315
5,504,921	4 1996	Dev et al.	709/223
5,511,197	4 1996	Hill et al.	709/328
5,524,244	6 1996	Robinson et al.	717/5
5,553,282	9 1996	Parrish et al.	707/10
5,555,367	9 1996	Premierani et al.	707/4
5,557,798	9 1996	Skeen et al.	395/650
5,569,063	9 1996	Nilsen et al.	
5,561,785	10 1996	Blandy et al.	395/497.01
5,577,231	11 1996	Scalzi et al.	703/26
5,603,031	2 1997	White et al.	395/683
5,617,537	4 1997	Yamada et al.	709/214
5,628,005	5 1997	Hurvig	707/8
5,640,564	6 1997	Hamilton et al.	709/315
5,644,768	7 1997	Periwal et al.	709/102
5,652,888	7 1997	Burgess	709/318
5,655,148	8 1997	Richman et al.	395/828
5,659,751	8 1997	Heninger	395/600
5,671,225	9 1997	Hoeper et al.	370/468
5,675,796	10 1997	Hodges et al.	709/100
5,680,273	10 1997	Rubin et al.	395/456
5,681,617	10 1997	Cough et al.	707/104
5,686,955	11 1997	Meyer et al.	709/316
5,689,709	11 1997	Corbett et al.	709/315
5,706,435	1 1998	Barbara et al.	395/200.09
5,724,588	3 1998	Hill et al.	709/328
5,727,145	3 1998	Nessett et al.	713/200
5,737,607	4 1998	Hamilton et al.	717/1
5,745,678	4 1998	Hertzberg et al.	713/200
5,745,695	4 1998	Gilchrist et al.	709/227
5,745,703	4 1998	Celjin et al.	709/238
5,754,849	5 1998	Dyer et al.	455/404
5,757,925	5 1998	Faybisenko	709/203
5,761,656	6 1998	Ben-Shachar	707/4
5,764,897	6 1998	Khalidi	709/201
5,768,532	6 1998	Megerian	709/245
5,774,551	6 1998	Wu et al.	713/155
5,778,228	7 1998	Wei	709/328
5,778,368	7 1998	Hogan et al.	707/10
5,787,425	7 1998	Bigus	707/6
5,787,431	7 1998	Shaughnessy	707/100
5,790,548	8 1998	Sistanizadeh et al.	370/401
5,809,507	9 1998	Cavanaugh, III	707/103
5,813,613	9 1998	Shakib et al.	707/102
5,815,149	9 1998	Mutschler, III et al.	345/335
5,815,709	9 1998	Waldo et al.	712/300
5,815,711	9 1998	Sakamoto et al.	717/1

5,829,022	10 1998	Watanabe et al.	711/118
5,832,529	11 1998	Wollrath et al.	707/206
5,832,593	11 1998	Wurst et al.	29/750
5,835,737	11 1998	Sand et al.	710/113
5,842,018	11 1998	Atkinson et al.	707/591
5,844,553	12 1998	Hao et al.	345/329
5,845,129	12 1998	Wendorf et al.	710/200
5,860,004	1 1999	Fowlow et al.	717/1
5,860,153	1 1999	Matena et al.	711/216
5,864,862	1 1999	Kriens et al.	707/103
5,864,866	1 1999	Henckel et al.	707/103
5,872,928	2 1999	Lewis et al.	709/222
5,875,335	2 1999	Beard	717/5
5,878,411	3 1999	Borroughs et al.	707/4
5,884,024	3 1999	Lim et al.	713/201
5,884,079	3 1999	Furusawa	717/1
5,887,134	3 1999	Ebrahim	709/200
5,890,158	3 1999	House et al.	707/10
5,892,904	4 1999	Atkinson et al.	713/201
5,933,497	8 1999	Beetcher et al.	705/59
5,935,249	8 1999	Stern et al.	713/201
5,940,827	8 1999	Hapner et al.	707/8
5,946,485	8 1999	Weeren et al.	717/3
5,946,694	8 1999	Copeland et al.	707/103
5,966,531	10 1999	Skeen et al.	709/315
6,003,763	12 1999	Gallagher et al.	235/379
6,009,103	12 1999	Woundy	370/401
6,061,699	5 2000	DiCecco et al.	707/513
6,061,713	5 2000	Bharadhwaj	709/203

FOREIGN PATENT DOCUMENTS

0 565 849	10/1993	(EP)	
0 569 195 A2	11/1993	(EP)	H04Q:3:545
0 569 195 A3	11/1993	(EP)	H04Q:3:545
0 625 750	11/1994	(EP)	
0 635 792	1/1995	(EP)	
0 651 328	5/1995	(EP)	
0 660 231	6/1995	(EP)	
0 697 655	2/1996	(EP)	
0 718 761	6/1996	(EP)	
0 767 432	4/1997	(EP)	
0 778 520	6/1997	(EP)	
0 794 493	9/1997	(EP)	
0 803 810	10/1997	(EP)	
0 803 811	10/1997	(EP)	
0 805 393	11/1997	(EP)	
0 810 524	12/1997	(EP)	
0 817 026	1/1998	(EP)	
0 817 022	1/1998	(EP)	
0 817 025	1/1998	(EP)	
0 836 140 A2	4/1998	(EP)	G06F:09:46
2 253 079	8/1992	(GB)	
2 262 825	6/1993	(GB)	
2 305 087	3/1997	(GB)	
11-45187	2/1996	(JP)	G06F:09:46
WO92/07335	4/1992	(WO)	
WO92/09948	6/1992	(WO)	
WO93/25962			
A1	12/1993	(WO)	
WO94/03855	2/1994	(WO)	
WO96/03692	2/1996	(WO)	
WO96/10787	4/1996	(WO)	
WO96/18947	6/1996	(WO)	
WO96/24099	8/1996	(WO)	
WO98/02814	1/1998	(WO)	
WO98/04971	2/1998	(WO)	

OTHER PUBLICATIONS

Mitchell, James G., et al., "Mesa Language Manual," Xerox Corporation, Palo Alto Research Centers.

- McDaniel, Gene. "An Analysis of a Mesa Instruction Set," Xerox Corporation, Palo Alto Research Centers, May 1982.
- Pier, Kenneth A., "A Retrospective on the Dorado, A High-Performance Personal Computer," Xerox Corporation, Palo Alto Research Centers, Aug. 1983.
- Pier, Kenneth A., "A Retrospective on the Dorado, A High-Performance Personal Computer," Conference Proceedings, The 10th Annual International Symposium on Computer Architecture, Royal Institute of Technology, Stockholm, Sweden, 1983.
- Birrell, Andrew, et al., "Distributed Garbage Collection for Network Objects," Digital Systems Research Center, Dec. 15, 1993.
- Gray, Cary G. and Cheriton, David R., "Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency," Computer Science Department, Stanford University, 1989.
- Hamilton, Marc A., "Java and the Shift to Net-Centric Computing," Computer, pp. 31-39, Aug. 1996.
- Birrell, Andrew, et al., "Network Objects," Digital Systems Research Center, Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles, vol. 27, No. 5, pp. 217-230, Dec. 1993.
- Mullender, *Distributed Systems*, Second Edition, Addison-Wesley, 1993.
- Howard et al., *Scale and Performance in a Distributed File System*, ACM Transactions on Computer Systems, vol. 6, No. 1, Feb. 1988, pp. 51-81.
- Cardelli, *Objig: A lightweight language for network objects*, Nov. 5, 1993, pp. 1-37.
- Dijkstra, *Self-stabilizing Systems in Spite of Distributed Control*, Communications of the ACM, vol. 17, No. 11, Nov. 1974, pp. 643-644.
- Ousterhout et al., "The Sprite Network Operating System," Computer, IEEE, Feb. 1988, pp. 23-26.
- Dourish, A Divergence-Based Model of Synchrony and Distribution in Collaborative Systems, Xerox Technical Report EPC-1, 94-102, 1994, pp. 1-10.
- Sharroff et al., ObjectMap: Integrating High Performance Resources into a Distributed Object-oriented Environment, ICODP, 1995.
- Birrell et al., Grapevine: An Exercise in Distributed Computing, Communications of the ACM, vol. 25, No. 4, Apr. 1982, pp. 260-274.
- Transparent Network Computing*, Locus Computing Corporation, Jan. 5, 1995.
- Lamport et al., The Byzantine Generals Problem, ACM Transactions on Programming Languages and Systems, vol. 4, No. 3, Jul. 1982, pp. 382-401.
- Dolev et al., On the Minimal Synchronism Needed for Distributed Consensus, Journal of the ACM, vol. 34, No. 1, Jan. 1987, pp. 77-97.
- Mumert et al., *Long Term Distributed File Reference Tracing: Implementation and Experience*, Carnegie Mellon University School of Computer Science, Nov. 1994, pp. 1-28.
- Gelemtier et al., *Parallel Programming in Linda*, Yale University, Jan. 1985, pp. 1-21.
- Cannon et al., Adding Fault-Tolerant Transaction Processing to Linda, Software-Practice and Experience, vol. 24(5), May 1994, pp. 449-466.
- Kambhata et al., Recovery with Limited Replay: Fault-Tolerant Processes in Linda, Oregon Graduate Institute, Technical Report CSIE 90-019, Sep. 1990, pp. 1-16.
- Anderson et al., Persistent Linda: Linda+Transactions+Query Processing Proceedings of the 13th Symposium on Fault Tolerant Systems, 1994, pp. 93-109.
- Gelemtier, Generative Communication in Linda, ACM Transactions on Programming Languages and Systems, vol. 7, No. 1, Jan. 1985, pp. 80-112.
- Carriero et al., Distributed Data Structures in Linda, Principles of Programming Language, 1986, pp. 1-16.
- Pinakis, *Using Linda as the Basis of an Operating System Microkernel*, University of Western Australia, Department of Computer Science, Aug. 1993, pp. 1-165.
- LINDA Database Search, Jul. 20, 1995, pp. 1-68.
- Carriero et al., *Distributed Data Structures in Linda*, Yale Research Report YALEU/DCS/RR-438, Nov. 1985.
- Agha et al., Actorspaces: An Open Distributed Programming Paradigm, University of Illinois, Report No. UIUCDCS-R-92-1766, Open Systems Laboratory TR No. 8, Nov. 1992, pp. 1-12.
- Ahmed et al., *A Program Building Tool for Parallel Applications*, Yale University, Dec. 1, 1993, pp. 1-23.
- Liskov et al., *Distributed Object Management in Thor*, International Workshop on Distributed Object Management, 1992, pp. 12.
- Coulouris et al., *Distributed Systems Concepts and Designs*, Second Edition, Addison-Wesley, 1994.
- Jaworski, *Java 1.1 Developer's Guide*, Sams.net, 1997.
- Wollrath et al., *A Distributed Object Model for the Java™ System*, USENIX Association, Conference on Object-Oriented Technologies and Systems, Jun. 17-21, 1996.
- Harris et al., *Proposal for a General Java Proxy Class for Distributed Systems and Other Uses*, Netscape Communications Corp., Jun. 25, 1997.
- Krasner, *The Smalltalk-80 Virtual Machine*, BYTE Publications Inc., Aug. 1991, pp. 300-320.
- Birrell et al., Network Objects, DEC SRC Research Report 115, Feb. 28, 1994.
- Remote Method Invocation Specification*, Sun Microsystems, Inc., (1997), <http://java.sun.com/products/jdk/1.1/docs/guide/rmi/spec/rmiTOC.doc.html>.
- Birrell, Andrew D. and Nelson, Bruce Jay, "Implementing Remote Procedure Calls," ACM Transactions on Computer Systems, vol. 2, No. 1, pp. 39-59, Feb. 1984.
- Wilson, P.R., et al., "Design of the Opportunistic Garbage Collector," Proceedings of the Object Oriented Programming Systems Languages An Applications Conference, New Orleans, vol. 24, No. 10, Oct. 1989.
- Jones, Richard, et al., "Garbage Collection: Algorithms for Automatic Dynamic Memory Management," John Wiley & Sons, Jan. 1996.
- Birrell, Andrew, et al., "Distributed Garbage Collection for Network Objects," Digital Systems Research Center, No. 116, Dec. 15, 1993, pp. 1-18.
- Gray, Cary G., et al., "Leases An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency," Computer Science Department, Stanford University, Jan. 1989, pp. 202-210.
- Birrell, Andrew, et al., "Network Objects," Digital Equipment Corp. Systems Research Center Technical Report, 27(5), Dec. 1993, pp. 217-230.
- Chung, Kin-Man and Yuen, Herbert, "A 'Tiny' Pascal Compiler: the P-Code Interpreter," BYTE Publications, Inc., Sep. 1978.

422

- Chung, Kin-Man and Yuen, Herbert, "A Tiny Pascal Compiler: Part 2: The P-Compiler," BYTE Publications, Inc., Oct. 1978.
- Thompson, Ken, "Regular Expression Search Algorithm," Communications of the ACM, vol. 11, No. 6, p. 149 et seq., Jun. 1968.
- Mitchell, James G., Maybury, William, and Sweet, Richard, Mesa Language Manual, Xerox Corporation, no date.
- McDaniel, Gene, "An Analysis of a Mesa Instruction Set," Xerox Corporation, May 1982.
- Pier, Kenneth A., "A Retrospective on the Dorando, A High-Performance Personal Computer," Xerox Corporation, Aug. 1983.
- Pier, Kenneth A., "A Retrospective on the Dorando, A High-Performance Personal Computer," IEEE Conference Proceedings, The 10th Annual International Symposium on Computer Architecture, Jan. 1983.
- K.Eric Drexler et al., "Incentive Engineering for Computational Resource Management," The Ecology of Computation, Elsevier Science Publishers B.V. (North Holland), Jan. 1988.
- IBM Technical Disclosure Bulletin, Nov. 1973, vol. 16, No. 6, p. 1931.
- Cary G. Gray and David R. Cheriton, "Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency," ACM (Association for Computing Machinery), pp. 202-210, 1989.*
- Alexander et al., "Active Bridging," Proceedings of the ACM SIGCOMM'97 Conference, Cannes, France, Sep. 1997.
- "Change-Notification Service for Shared Files," IBM Technical Disclosure Bulletin, vol. 36, No. 8, pp. 77-82, NP002108713, New York, US, Aug. 1993.
- Beech, et al., "Object Databases as Generalizations of Relational Databases," Computer Standards & Interfaces, vol. 13, Nos. 1-3, pp. 221-230, Amsterdam, NL, Jan. 1991.
- Bertino et al., "Object-Oriented Database Management Systems: Concepts and Issues," Computer, vol. 24, No. 4, pp. 33-47, Los Alamitos, CA, Apr. 1991.
- Betz, "Interoperable Objects: Laying the Foundation for Distributed Object Computing," Dr. Dobbs' Journal, vol. 19, No. 11, p. 18(13), Oct. 1994.
- Bevan, "An Efficient Reference Counting Solution To The Distributed Garbage Collection Problem," Parallel Computing, NL, Elsevier Science Publishers, Amsterdam, vol. 9, No. 2, pp. 179-192, Jan. 1989.
- Dave et al., "Proxies, Application Interface, And Distributed Systems," Proceedings International Workshop On Object Orientation In Operating Systems, pp. 212-220, Sep. 24, 1992.
- Deux et al., "The O2 System," Communications Of The Association For Computing Machinery, vol. 34, No. 10, pp. 34-48, Oct. 1, 1991.
- Droms, "RFC 1541 Dynamic Host Configuration Protocol," <http://www.cis.ohio-state.edu/hbin/rfc/rfc1541.html>, pp. 1-33, Oct. 1993.
- Lums, "A Definition Of An Access Control Systems Language," Computer Standards And Interfaces, vol. 6, No. 4, pp. 443-454, Jan. 1, 1987.
- Gosling et al., "Java (TM) Language Specification," Addison-Wesley, Jan. 1996.
- Gottlieb et al., "Extending Object-Oriented Systems with Roles," ACM Transactions On Information Systems, vol. 14, No. 3, pp. 268-296, Jul. 1996.
- Guth, "JavaOne: Sun to Expand Java Distributed Computing Effort," <http://www.sunworld.com/swol-02-1998/swol-02-sunspots.html>, XP-002109935, p. 1, Feb. 20, 1998.
- Hamilton et al., "Subcontract: A Flexible Base For Distributed Programming," Proceedings of 14th Symposium of Operating System Principles, Dec. 1993.
- Hunt, "IDF: A Graphical Data Flow Programming Language for Image Processing and Computer Vision," Proceedings of the International Conference on Systems, Man, and Cybernetics, pp. 351-360, Los Angeles, Nov. 4-7, 1990.
- IBM (TM) Technical Disclosure Bulletin, "Object Location Algorithm," vol. 36, No. 09B, pp. 257-258, Sep. 1993.
- IBM (TM) Technical Disclosure Bulletin, "Retrieval of Qualified Variables Using Extendible Hashing," vol. 36, No. 12, pp. 301-303, Dec. 1993.
- IBM, "Chapter 6—Distributed SOM (DSOM)," SOMobjects Developer Toolkit Users Guide, Version 2.1, p. 6-1-6-90, Oct. 1994.
- Kay et al., "An Overview of the Raleigh Object-Oriented Database System," ICL Technical Journal, vol. 7, No. 4, pp. 780-798, Oxford, GB, Nov. 1991.
- Kougiouris et al., "Support for Space Efficient Object Invocation in Spring," Sep. 1994.
- Lindholm et al., "The Java (TM) Virtual Machine Specification," Addison Wesley, Jan. 1996.
- Mitchell et al., "An Overview of the Spring System," Feb. 1994.
- Orfali et al., "The Essential Distributed Objects Survival Guide," Chapter 11: Corba Commercial ORBs, pp. 203-215, John Wiley & Sons, Inc., Jan. 1996.
- Riggs et al., "Pickling State in the Java (TM) System," USENIX Association Conference on Object-Oriented Technologies and Systems, XP-002112719, pp. 241-250, Jun. 17-21, 1996.
- Rosenberry et al., "Understanding DCE," Chapters 1-3, 6, 1992.
- Waldo et al., "Events In An RPC Based Distributed System," Proceedings Of The 1995 USENIX Technical Conference, Proceedings USENIX Winter 1995 Technical Conference, New Orleans, LA, USA, 16-20, pp. 131-142, Jan. 1995.
- Wu, "A Type System For An Object-Oriented Database System," Proceedings of the International Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, pp. 333-338, Sep. 11-13, 1991.
- Yemini et al., "Towards Programmable Networks," IFIP: IEEE International Workshop on Distributed Systems: Operations and Management, L'Aquila, Italy, Oct. 1996.
- Aldrich et al., "Providing Easier Access to Remote Objects in Client-Server Systems," System Sciences, 1998, Proceedings of the 31st Hawaii International Conference, Jan. 6-9, 1998, pp. 366-375.
- Aldrich et al., "Providing Easier Access to Remote Objects in Distributed Systems," Calif. Institute of Technology, www.cs.caltech.edu/%7Ejedi/paper/jedipaper.html, Nov. 21, 1997.
- Dave et al., "Proxies, Application Interfaces, and Distributed Systems," XP 002009478, IEEE, Jan. 1992, pp. 212-220.
- Dollimore et al., "The Design of a System for Distributing Shared Objects," The Computer Journal, No. 6, Cambridge, GB, Dec. 1991.

424

Fleisch et al., "High Performance Distributed Objects Using Distributed Shared Memory & Remote Method Invocation," System Sciences, 1998, Proceedings of the 31st Hawaii Internat'l. Conference, Jan. 6-9, 1998, pp. 574-578.
 Guyennet et al., "A New Consistency Protocol Implemented in the CAH System," IEEE, 1094-7256/97, pp. 82-87, Jan. 1997.
 Guyennet et al., "Distributed Shared Memory Layer for Cooperative Work Applications," IEEE, 0742-1303/97, pp. 72-78, Jan. 1997.

IBM Technical Disclosure Bulletin, "Local Network Monitoring to Populate Access Agent Directory," vol. 36, No. 09A, pp. 403-405, Sep. 1993.
 MUX-Elektronik, Java 1.1 Interactive Course, www.ils.se/~mux/javaic.html, Jan. 1995.
 Stevenson, "Token-Based Consistency of Replicated Servers," IEEE, CH2686-4/89/0000/0179, pp. 179-183, Jan. 1989.

* cited by examiner

10012). For example, if access to the delegator's tax files are requested, the delegator may grant a lease period up to April 15. Next, the delegator creates a lease object and returns the object to the delegatee (step 10020) and processing ends.

FIGS. 11A and 11B depict a flowchart of the steps performed by the delegatee 9010 when requesting a lease from the delegator 9008. The first step performed by the potential delegatee is to receive a request by the delegatee that entails accessing a file on the delegator's behalf (step 11001). At some point later, the delegatee sends a request for a lease to the delegator (step 11002). This request is a function call and is described in step 10004 in FIG. 10.

After sending the request, the delegatee receives a lease object from the delegator (step 11004). The lease object, as described above, includes the principal identifier, the authentication method, the access-type method, the lease duration method, the renew method, the cancel method.

Next, the delegatee, by examining the lease object, determines if a lease was granted (step 11006). If the delegatee determines that a lease was not granted, the delegatee invokes the exception method (step 11008), which allows the delegatee to determine why a lease was not granted. If the delegatee determines that the lease was not granted because of an improper request (step 11010), the delegates reconfigures the request (step 11012), and processing continues to step 11002. However, if the delegatee determines that the lease was not granted for reasons other than an improper request, processing ends. Note, the delegator may grant multiple leases to the same file, since it is ultimately the responsibility of the file system manager to referee actual access to the file.

If the delegatee, in step 11006, determines that a lease was granted by the delegator, next the delegates requests access to the file from the file system manager, by sending an access request (step 11018). Processing then continues to step 11020 in FIG. 11B.

After the delegatee sends the access request to the file system manager in step 11018 in FIG. 11A, the delegatee determines, by examining the lease object, if the file system manager granted the delegatee access to the file (step 11020). If the delegates determines that access was not granted by the file system manager, the delegatee invokes the exception method, contained in the object returned by the file system manager, which allows the delegatee to determine why access was not granted (step 11022).

If the delegatee determines that access was not granted because of an improper request (step 11024), processing ends. On the other hand, if the request was proper, the delegatee determines if access was not granted because the file system manager allocated the file to another leaseholder (step 11026). If the delegatee determines the file is busy, the delegatee waits for a predetermined period of time (step 11028) and processing continues to step 11018 in FIG. 11A. If the delegatee determines that access to the file was denied for some other reason, processing ends.

If the delegatee determines, in step 11020, that the file system manager granted the delegates access to the file, then the delegates can access the file (step 11030). Next, the delegatee determines if it is finished accessing the file (step 11032).

If the delegatee's use is completed, the delegatee determines if the lease expired, (i.e., the lease is no longer active) (step 11034). If the lease expired, processing ends and no communication is necessary between the delegatee and the file system manager (i.e., natural termination occurs). Otherwise, if the lease is still active the delegatee invokes the cancel method (step 11036). The delegatee accesses the

cancel method via the lease object. The cancel method informs the file system manager and the delegator that the delegatee is no longer interested in the file. Accordingly, the cancel method allows the file system manager to reclaim the file for use by other lease holders in an expeditious fashion.

If the delegatee determines in step 11032 that it still desires access to the file, the delegatee determines if the lease is about to expire (step 11038). This is achieved by the delegatee comparing the duration of the lease with current time minus the time when the lease was granted. The duration of the lease is found by invoking the duration method. If the lease is not about to expire, the delegatee continues to access the file (step 11030).

However, if the lease is about to expire in step 11038, the delegatee must decide whether or not to renew the lease (step 11040). If the delegatee chooses to renew the lease, the delegatee invokes the renew method of the lease object. If the renew method is invoked, processing continues to step 11002 in FIG. 11A. If the delegates does not renew the lease, then processing ends and no communication is necessary between the delegatee and the file system manager (i.e., natural termination occurs).

FIG. 12 depicts a flow chart of the steps performed by the file system manager 9016 when a delegatee process 9008 requests access to a file. The first step performed by the file system manager is to receive an access request by the delegatee (step 1202). After receiving the request, the file system manager authenticates the delegatee's identity by invoking the principal identifier method and the authentication method, via the lease object (step 1203). If the file system manager determines that the delegatee's lacks the proper identity to access the file (step 1204), the file system manager generates an exception (step 1206) and processing ends.

If the file system manager determines in step 1204 that the delegatee's identity is authentic, the file system manager invokes the getAccess type method (step 1208). By invoking the getAccess type method, the file system manager is able to determine which type of access the delegatee desires. Next, the file system manager determines if the file is available for the type of access requested (step 1210). If the file system manager determines that the file is unavailable, the file system manager generates an exception (step 1206) and processing ends.

If the file system manager determines, in step 1210, that the file is available, the file system manager invokes the duration method (step 1212). The file system manager invokes the duration method in order to determine if there is time left on the delegatee's lease. If the file system manager determines that the delegatee's lease is active (step 1214), the file system manager grants the delegatee access to the file (step 1218). After granting the delegatee access to the file, the file system manager returns to step 1212.

If the file system manager determines in step 1214 that the lease is not active, the file system manager will reclaim the file (step 1216). After reclaiming the file, the file system manager generates an exception (step 1206) and processing ends.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. The scope of the invention is defined by the claims and their equivalents.

What is claimed is:

1. A method in a processing system, comprising the steps of:

17

receiving a lease request from a program, the lease request specifying a resource and a requested lease period;
determining a lease period during which the program has authority to access the resource; and
sending to the program a delegation certificate for use by the program to access the resource during the determined lease period.

2. A method for managing a resource in a system that includes a client, a server that manages the resource, and a delegator authorized to use the resource, said method comprising the steps of:

the client receiving from the delegator a lease that delegates to the client use of the resource for a lease period; and
the server providing the client with access to the resource for the lease period.

3. The method of claim 2, wherein the receiving step includes the step of:

requesting by the client a lease from the delegator for using the resource.

4. The method of claim 2, wherein the receiving step includes the step of:

providing by the delegator to the client a delegation certificate that delegates to the client the delegator's authority to use the resource.

5. The method of claim 2, wherein the providing step includes the step of:

identifying an identity of the client by invoking code provided by the delegator to the client.

6. The method of claim 2, wherein the providing step includes the step of:

authenticating an identity of the client by invoking code provided by the delegator to the client.

7. The method of claim 2, wherein the providing step includes the step of:

the client managing the lease by invoking code provided by the delegator to the client.

8. A method for managing a resource in a system that includes a client, a server that manages the resource, and a delegator authorized to use the resource, said method comprising the steps of:

the client requesting from the delegator a lease for using the resource;

the delegator granting the requested lease to the client for a lease period; and

the client using the resource for the lease period.

9. The method of claim 8, wherein the granting step includes the step of:

determining the lease period during which the client is authorized to use the resource.

10. The method of claim 8, wherein the granting step includes the step of:

providing to the client a delegation certificate that authorizes the client to use the resource for the granted lease period.

11. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for identifying the client as the delegator to the server.

12. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for use by the server for authenticating identification of the client.

13. The method of claim 12, wherein the providing step includes the step of:

18

providing to the client an authentication method that uses a public key.

14. The method of claim 12, wherein the providing step includes the step of:

providing to the client an authentication method that uses a challenge-response protocol.

15. The method of claim 12, wherein the providing step includes the step of:

providing to the client an authentication method that uses a shared secret method.

16. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for determining a type of access to the resource authorized by the delegator.

17. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for use by the server for determining a type of access that the server grants to the client for using the resource when the client requests from the server access to the resource.

18. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for use by the server for determining the lease period.

19. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for canceling the granted lease when the client completes using the resource.

20. The method of claim 8, wherein the granting step includes the step of:

providing to the client a method for renewing the granted lease when the lease period is near expiration.

21. The method of claim 8, wherein the using step includes the step of:

the client requesting from the server access to the resource according to the granted lease.

22. The method of claim 8, further comprising the step of:

reclaiming the resource by the server when the client cancels the granted lease.

23. The method of claim 8, further comprising the step of:

reclaiming the resource by the server when the granted lease expires.

24. The method of claim 8, further comprising the step of:

the delegator requesting from the client to perform a task that requires use of the resource.

25. The method of claim 8, wherein the using step includes the step of:

the client requesting from the server access to the resource.

26. The method of claim 8, wherein the using step includes the step of:

authenticating by the server, based on a delegation certificate provided by the delegator, an identity of the client when the server receives a request from the client to use the resource.

27. The method of claim 8, wherein the using step includes the step of:

the server determining, based on a delegation certificate provided by the delegator, a type of access by the client to the resource.

28. A data processing system, comprising:

a memory including:

a client program containing first code that requests a lease for accessing a resource;

19

a delegator program having access to the resource and containing second code for granting the lease to the client for a lease period by providing the client program with a delegation certificate; and
 a server program containing third code for granting access by the client to the resource based on the delegation certificate; and
 a processor for running the client program, the server program, and the delegator program.

29. An apparatus, comprising:
 a requesting means for requesting access to a resource;
 a delegation means for delegating to the requesting means an authorization to access the resource; and
 a resource allocating means for granting access by the requesting means to the resource based on the delegated authorization.

30. A computer-readable memory device encoded with a data structure for using a resource in a system including a client, a server that manages the resource, and a delegator authorized to access the resource, the data structure comprising:
 an object including:
 a first method for identifying the client to the server as the delegator when the client requests from the server access to the resource;
 a second method for authenticating identification of the client;
 a third method for determining type of access by the client to the resource and duration of access by the client to the resource; and
 a fourth method for managing access by the client to the resource.

31. The computer-readable memory device of claim 30, wherein the fourth method includes a method for renewing a lease granted by the delegator to the client for accessing the resource.

32. The computer-readable memory device of claim 30, wherein the fourth method includes a method for canceling a lease granted by the delegator to the client for accessing the resource.

33. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system including a client, a server that manages a resource, and a delegator authorized to access the resource, said method comprising the steps of:
 the client requesting from the delegator a lease for using the resource;
 the delegator providing to the client a delegation certificate that grants the lease to the client for a lease period; and
 the server granting access to the resource by the client based on the delegation certificate.

34. The computer-readable medium of claim 33, further comprising the step of:
 managing by the client the granted lease using the delegation certificate.

35. The computer-readable medium of claim 33, wherein the providing step includes the step of:
 providing to the client a method for determining duration of the granted lease period.

36. The computer-readable medium of claim 33, wherein the providing step includes the step of:
 providing to the client a method for determining a type of access authorized by the delegator.

37. The computer-readable medium of claim 33, wherein the providing step includes the step of:
 providing to the client a method for renewing the granted lease.

20

38. The computer-readable medium of claim 33, wherein the providing step includes the step of:
 providing to the client a method for canceling the granted lease.

39. The computer-readable medium of claim 33, wherein the granting step includes the step of:
 invoking a method in the delegation certificate to determine identity of the client.

40. The computer-readable medium of claim 33, wherein the granting step includes the step of:
 invoking a method in the delegation certificate to authenticate identification of the client.

41. The computer-readable medium of claim 33, wherein the granting step includes the step of:
 invoking a method in the delegation certificate to determine a type of access by the client to the resource.

42. The computer-readable medium of claim 33, wherein the granting step includes the step of:
 invoking a method in the delegation certificate to determine duration of the granted lease period.

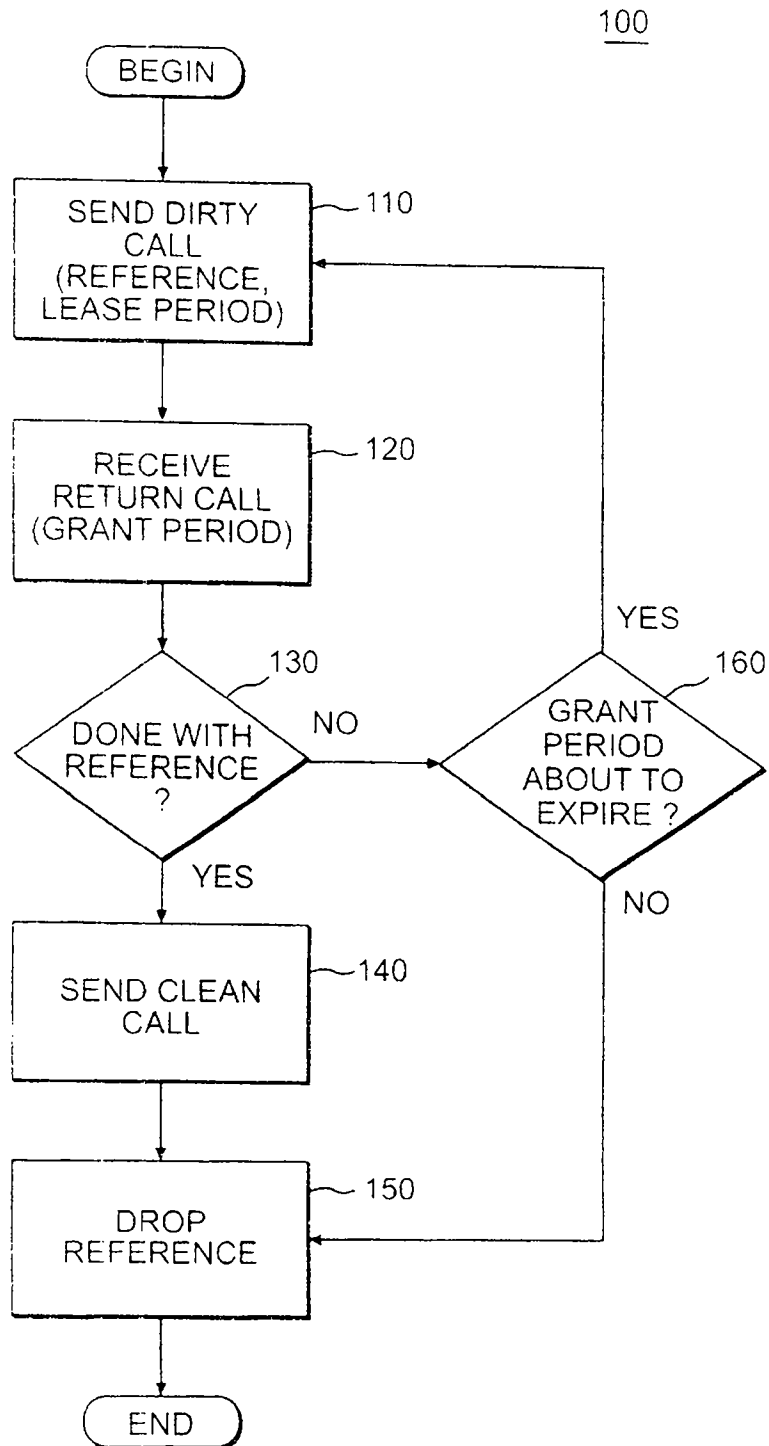
43. A method for using a resource in a system that includes a client, a server that manages the resource, and a delegator authorized to use the resource, said method comprising the steps of:
 the client requesting from the delegator a lease for accessing the resource;
 the client receiving from the delegator a lease object that delegates an authorization to use the resource for a lease period; and
 the client accessing the resource according to the delegated authorization.

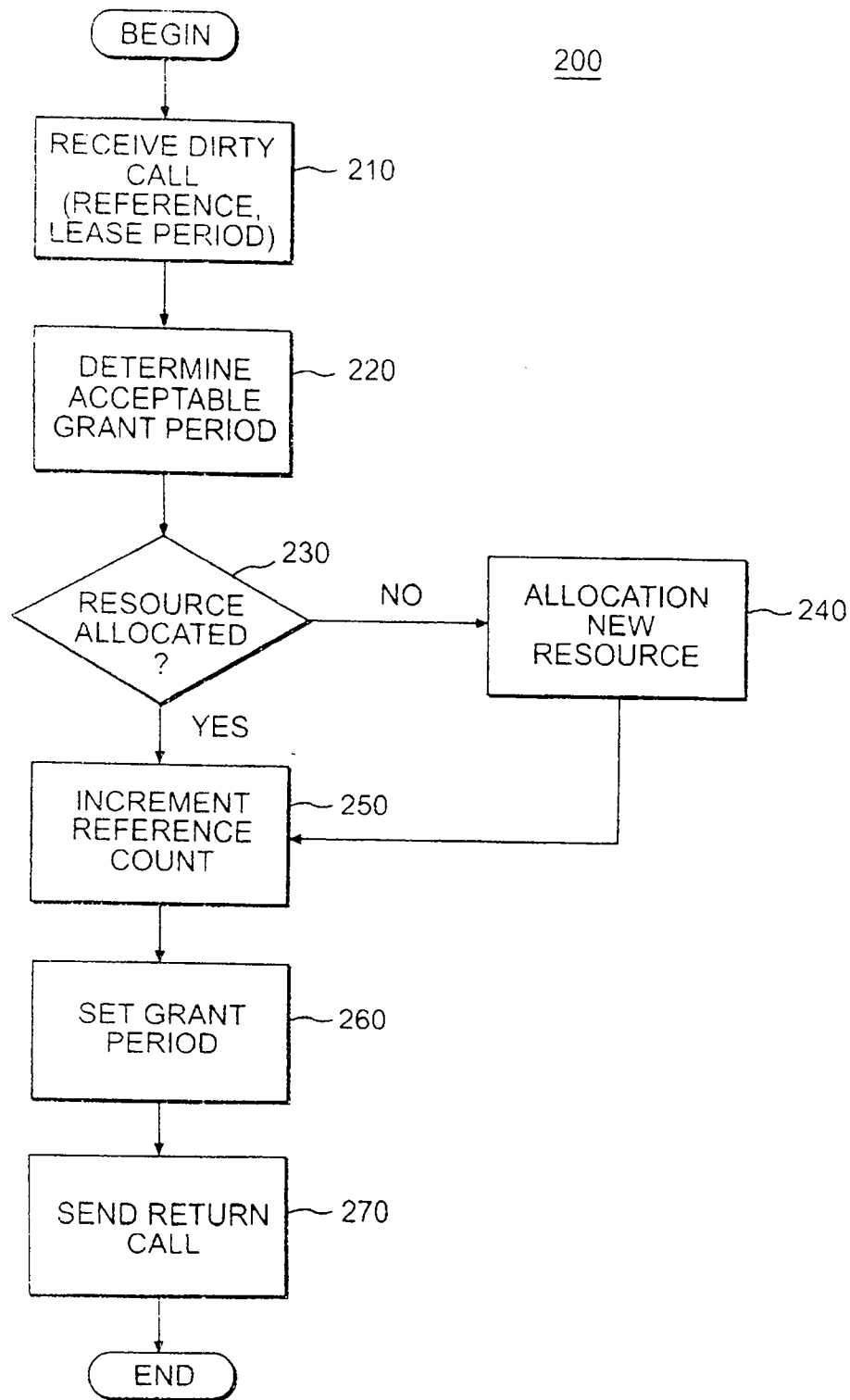
44. The method of claim 43, further comprising the step of:
 the client managing the lease by invoking one or more methods in the lease object.

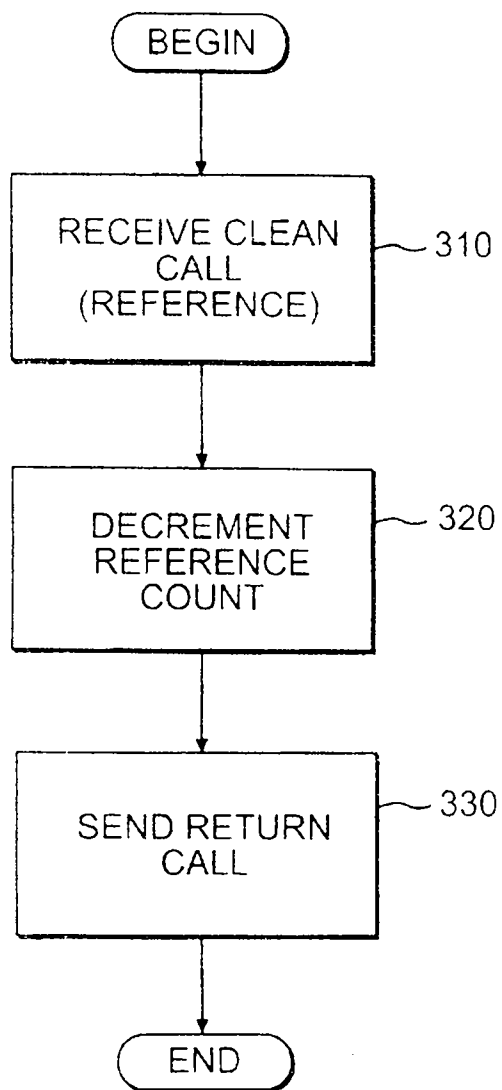
45. A method for managing a resource in a system that includes a client, a server that manages the resource, and a delegator authorized to use the resource, said method comprising the steps of:
 the delegator receiving from the client a request for a lease for accessing the resource; and
 the delegator providing a lease object that delegates to the client an authorization to use the resource for a lease period.

46. A method for managing a resource in a system that includes a client, a server that manages the resource, and a delegator authorized to use the resource, said method comprising the steps of:
 the server receiving from the client a request for accessing the resource, the request having an associated lease object that delegates to the client an authorization from the delegator to access the resource for a lease period;
 the server authenticating the request by invoking a first method on the lease object;
 the server determining the lease period by invoking a second method on the lease object;
 the server determining a type of access by the client to the resource by invoking a third method on the lease object; and
 the server providing the client with access to the resource when the request is authenticated, the lease period is active, and the type of access is authorized.

* * * * *

**FIG. 1**



300**FIG. 3**

400

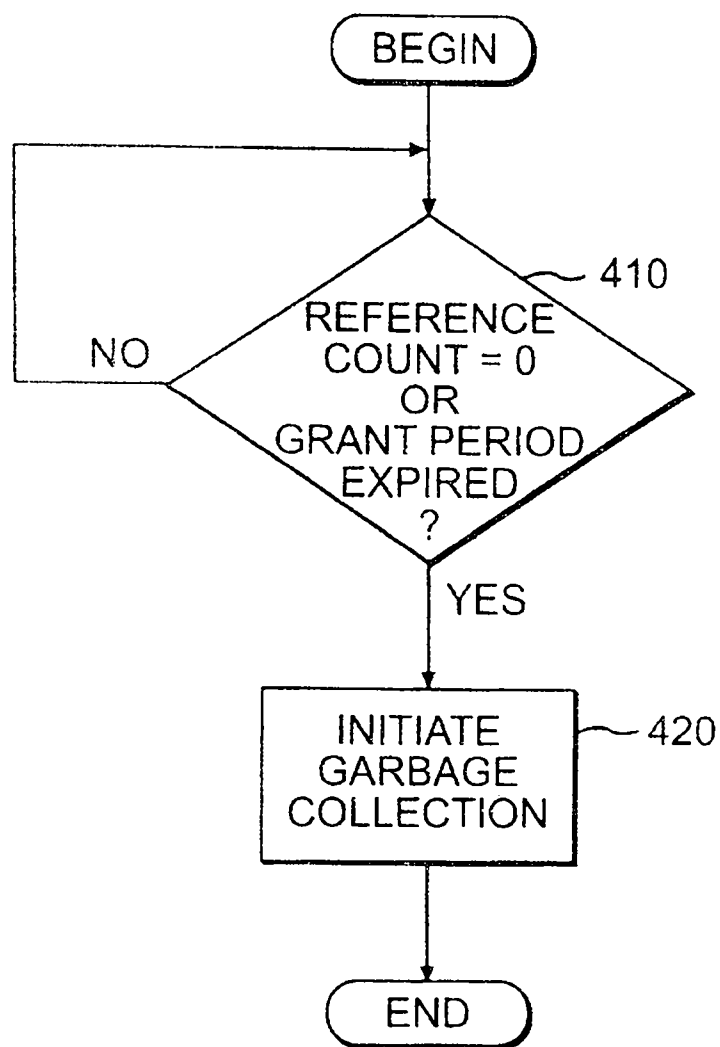


FIG. 4

(19)  **Europäisches Patentamt**
European Patent Office
Office européen des brevets



(11) **EP 0 952 513 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
 27.10.1999 Bulletin 1999/43

(51) Int. Cl.⁶: **G06F 3/12**

(21) Application number: **98118576.2**

(22) Date of filing: **01.10.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
 Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
 • **White, Craig R.**
 Eagle, ID 83616 (US)
 • **Albright, Shivaun**
 Rocklin, CA 95677 (US)

(30) Priority: **24.04.1998 US 66088**

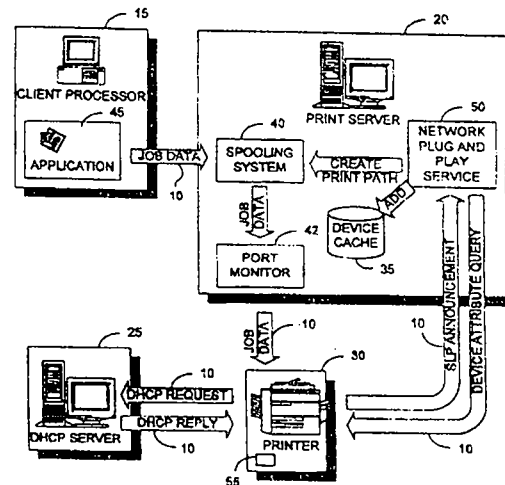
(74) Representative:
Schoppe, Fritz, Dipl.-Ing.
Schoppe, Zimmermann & Stöckeler
 Patentanwälte
 Postfach 71 08 67
 81458 München (DE)

(71) Applicant:
Hewlett-Packard Company
 Palo Alto, California 94304 (US)

(54) **Automatic configuration of a network printer**

(57) A method of configuring a peripheral device (30) on a network (10) without user intervention includes a server node (20) receiving identification data (130) for identifying the peripheral device on the network and, if the peripheral device is a new device (135) on the network or a driver for the peripheral device is not registered (145) on the server node, the server node self-installing the driver (160) for the peripheral device. As such, newly connected peripheral devices are automatically seen on a network and a driver for the peripheral device is automatically configured on a print server without any user intervention. Additionally, a communication port and queue for the device are also automatically created (175,185). Moreover, device configurations are dynamically and automatically updated (190,192,195) on the print server for consistent, accurate network information.

FIG. 1



EP 0 952 513 A1

Description

FIELD OF THE INVENTION

[0001] This invention relates in general to network communications and, more particularly, to enabling network peripheral devices such as printers to be automatically configured without user intervention upon connection to a network.

BACKGROUND OF THE INVENTION

[0002] Enabling a user to employ a newly added printer over a network is often a complex task. At an overview level, initial set-up in current network environments generally requires the creation of print queues, printer objects and print servers. Then, such queues, objects and print servers must be properly linked and configured. Then, to send a print job to the network printer, the client processor usually directs the print job to a network queue on the print server, and then the print server processes the job from the queue and sends it to the printer.

[0003] More specifically, adding a printer to a network environment is, conventionally, no small task. It requires a series of installation and configuration steps that are generally performed either by a user or a network administrator or both in cooperation with each other. Namely, the printer requires specific configuration steps to be performed and parameters to be applied for successful operation on the network, as also does any computer (client processor) on the network that will be sending data to the printer. For example, typically, a vendor's configuration application program must be executed on a client computer to identify the new printer (device) to be used on the network. Once the new printer is identified from a list of devices presented by the vendor's configuration application, then the configuration application provides proper configuration parameters for that new device. The configuration parameters are very specific to the device being added and to the client computer and network over which communication occurs.

[0004] Additionally, a print server must be configured to enable use of the newly added printer on the network. Conventionally, the print server manages and monitors use of the printer relative to all client computers on the network, and all client computers access the printer via the print server. Accordingly, the print server must also be configured, in addition to the client computer, to enable use of the printer over the network. This includes, typically, a vendor's configuration application also being executed on the print server for that printer.

[0005] In addition to a vendor's configuration application being executed to install and configure the printer on the print server, the network operating system must also execute its configuration routines on the print server in order to configure and enable the printer for

the network via the print server. This often includes identifying appropriate printer files, registering the printer with the spooling system, establishing printer queues and printer ports, and defining client processor network share-access rights to the printer. Unfortunately, the vendor configuration application and the network operating system configuration routines are generally not integrated, and must also be accessed directly from the print server itself. Thus, a network administrator conventionally manually initiates this software configuration process for the newly added printer from the print server itself.

[0006] While such a conventional network printer configuration system and method works well, when properly configured, it is more complex and cumbersome than what most users care to engage in. In addition, if the installation or configuration doesn't work quite right, troubleshooting can be both difficult and time-consuming.

[0007] Accordingly, an object of the present invention is to enable a peripheral device, such as a printer, to be automatically configured without user intervention upon connection to a network.

SUMMARY OF THE INVENTION

[0008] According to principles of the present invention in a preferred embodiment, a method of configuring a peripheral device on a network without user intervention includes the server node receiving identification data for identifying the peripheral device on the network and, if the peripheral device is a new device on the network or a driver for the peripheral device is not registered on the server node, the server node self-installing the driver for the peripheral device.

[0009] Thus, newly connected peripheral devices are automatically seen on a network and a driver for the peripheral device is automatically configured on a server node without any user intervention. Additionally, a communication port and queue for the device are also automatically created. Moreover, device configurations are dynamically and automatically updated on the server for consistent, accurate network information.

[0010] Other objects, advantages, and capabilities of the present invention will become more apparent as the description proceeds.

DESCRIPTION OF THE DRAWINGS

[0011]

FIG. 1 is a block diagram of a communication network and attached devices and includes the present invention for automatically configuring a network printer.

FIGs. 2-3 are flow charts depicting a preferred method of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0012] FIG. 1 is a high level block diagram of a communication network 10 and attached devices including a client processor 15, print server 20, DHCP server 25, and network printer 30. The present invention is described in the context of communication network 10 being a conventional TCP/IP network. In the diagram, network 10 is illustrated as directional arrows to represent the TCP/IP communication links and data flow paths between each of the devices over the network in association with the present invention. The present invention is further described herein in the context of client processor 15, print server 20 and DHCP server 25 employing the Windows[®] or Windows NT[®] operating system produced by Microsoft Corporation. However, the present invention is also applicable to other networking environments and operating systems. Additionally, although a laser printer 30 is disclosed herein for purposes of discussion, it is understood that it is merely exemplary and that the present invention is equally applicable to other network peripheral devices, including ink jet printers, digital copiers, facsimile machines, etc. Moreover, each device 15, 20 and 25 includes a conventional network input/output (I/O) card for enabling communications with network 10. Printer 30 also includes a "conventional" network I/O card for general network communications purposes, but it is modified per the present invention as will be discussed hereafter.

[0013] Print server 20 includes device cache memory 35, spooling system 40, and port monitor 42 as conventional in the art. Client processor 15 includes application software 45 as conventional in the art for sending a print job to printer 30. DHCP server 25 is a Dynamic Host Configuration Protocol (DHCP) server that holds TCP/IP configuration parameters available for printer 30 and/or other devices to be configured on network 10. DHCP server 25 is shown and discussed as a device that is physically separate from print server 20 for clarity of discussion and illustration. However, DHCP server 25 and print server 20 may be a single device having the separate functionalities combined therein.

[0014] Overall, the present invention enables printer 30 to be automatically configured on network 10 for use by client processor 15 after it (the printer) is physically connected to the network by a user. Specifically, "network plug and play" software 50 depicts a key component of the present invention and resides in a memory on print server 20. A coexisting "network plug and play" module 55 resides (as firmware) on the network I/O card in printer 30. Thus, the network I/O card on printer 30 is modified from a conventional network I/O card in that it includes the "network plug and play" module 55 of the present invention. These software routines (50,55) execute and interact to enable the automatic configuration of printer 20 without any user intervention after the printer is connected to the network.

[0015] Referring now to FIGs. 2-3 (in conjunction with

FIG. 1), a flow chart describes the functionality and a preferred method of the present invention. First, 105 (FIG. 2), printer 30 is connected to the network 10 and turned on. Upon power up, printer 30 (module 55) issues a DHCP request packet 110 over network 10. DHCP server 25 responds with a DHCP reply packet 115 that contains the available TCP/IP configuration parameters for printer 30. Upon receiving those TCP/IP configuration parameters 120, printer 30 sets its memory configuration to reflect the parameters received. The printer now has its TCP/IP address.

[0016] Next, 125, module 55 of printer 30 sends a Service Location Protocol (SLP) reply packet using IP multicast over network 10. This SLP reply packet serves as an announcement that the printer is up and running on the network. The SLP reply packet contains information specific to printer 30, including the printer's hardware (network I/O card) address, TCP/IP address, printer model name, network I/O card name, and TCP/IP hostname.

[0017] When "network plug and play" service 50 receives the SLP packet data 130, it initiates a search of device cache 35 on print server 20 to determine whether printer 30 (as identified in the SLP reply packet data) is a new device on the network. If printer 30 is a new device 135 on the network (i.e., there is not an entry for printer 30 in cache 35), then it is added as a new entry 140 to cache 35. Additionally, and importantly, after adding printer 30 as a new entry to cache 35, "network plug and play" service 50 then installs the appropriate print driver for printer 35 onto printer server 20 based on the printer model provided in the SLP reply packet.

[0018] Automatically installing the appropriate print driver includes communicating 145 (FIG. 3) with spooling system 40 to see if the appropriate print driver is already registered therein. If it is not, the installation source (location) of the network operating system is determined 150 (i.e., from a saved location in the Windows and Windows NT operating systems) and then that installation source (i.e., local or remote disk drive) is accessed in an attempt to automatically obtain the print driver. If the original disk that was used to install the network operating system is available at the source location 155 (i.e., in the drive), then the appropriate print driver for printer 30 is accessed 160 and registered with spooling system 40. Thus, the print driver installation is completely automated by "network plug and play" software 50, 55 of the present invention.

[0019] Only if the original disk does not exist at the determined source location 150, 155 will a user be prompted for input 165 to identify the location of the installation disk. However, this prompting only occurs once because the "network plug and play" module subsequently stores 170 the user input source location. As such, upon a next attempt to install another printer, the newly identified installation location is automatically accessed for obtaining the appropriate print driver.

Then, only if the installation disk is not in that location will another prompt occur for user input to identify a new installation location.

[0020] Subsequent to accessing the print driver 150, 155, 160 and registering it with spooling system 40, "network plug and play" software 50 then creates a communication port 175 on print server 20 using the hostname from the SLP reply packet or a generated port name based on the hardware address of the device. Additionally, a print queue is created 180 on print server 20 using a unique name based on the model name of printer 30 (supplied in the SLP reply packet). The print queue is then designated with network share access rights 180 so client processor 15 can use printer 30 over network 10.

[0021] Finally, after basic communication and configuration parameters have been established, "network plug and play" software 50 then queries printer 30 for additional attribute information 185, such as how much memory the printer has, text describing the printer, number of trays, print resolution, etc. This information is then sent to spooling system 40 to allow client processor 15 to locate printer 30 based on such attributes. At this point, client processor 15 is now able to send print data to printer 30 via print server 20.

[0022] Referring back now for a moment to the determination 130, 135 (FIG. 2) of whether an entry already exists for printer 30 in cache 35, if an entry exists (i.e., the printer is not a new device on the network), then no change is made in cache 35. However, if some configuration parameters have changed 190 for printer 30 (as detected in the SLP reply packet information), but the printer model has not changed 192, then the spooler 40 configuration parameters are adjusted accordingly 195. For example, if an I/P address for printer 30 has changed, then the port address in spooling system 40 is modified accordingly.

[0023] On the other hand, if the printer model has changed 192, then a determination is made 145 (FIG. 3) as to whether the print driver for that new (changed) model is already registered in the spooler. If the driver for the changed model is not already registered 145, then steps are taken 150, 155, 160 to access and install the appropriate driver (as previously discussed). If the driver is already registered 145, then we simply ensure that a proper communication port 175 and print queue 180 are created, and we verify attribute information 185.

[0024] In either case, whether certain printer parameters are noted to have changed, or whether the printer model itself has changed, these processes discussed for capturing the changes represent a unique, dynamic updating capability of the present invention for automatically maintaining accurate network information for printer 30 on print server 20 without any user intervention.

[0025] Advantageously, the present invention enables new peripheral devices (such as printer 30) to be automatically seen on a network with the only user input

requirement being to physically connect the printer to the network. Importantly, the printer driver is automatically configured on print server 20 without any user intervention, the port and queue for the printer are likewise automatically created, and printer configurations are dynamically and automatically updated for consistent, accurate network information.

[0026] In summary, the present invention provides an automatic configuration of a network peripheral device without user intervention. It will be obvious to one of ordinary skill in the art that the present invention is easily implemented utilizing any of a variety of components and tools existing in the art. Moreover, while the present invention has been described by reference to specific embodiments, it will be apparent that other alternative embodiments and methods of implementation or modification may be employed without departing from the true spirit and scope of the invention.

Claims

1. A method of configuring a peripheral device (30) on a network (10) without user intervention, the network having a server node (20) connected thereto, the method comprising:
 - (a) the server node (20) receiving identification data (130) for identifying the peripheral device (30) on the network (10); and,
 - (b) upon detecting the received identification data, the server node self-installing a driver (160) for the peripheral device if the driver is not already installed (145) on the server node (20).
2. The method of claim 1 wherein the peripheral device (30) is selected from a printing device, digital copying device, or facsimile device.
3. The method of claim 1 or 2 wherein the step of self-installing the driver (160) includes accessing a driver file for the peripheral device and registering the driver file with the server node.
4. The method of claim 3 wherein accessing the driver file includes accessing a storage device (150,155) identified from a memory location in the server node (20), the storage device being identified as a source from which the server node was network enabled.
5. The method of claim 1, 2, 3 or 4 wherein the peripheral device is a printer (30) and wherein the step of self-installing the driver (160) includes registering the peripheral device with a spooling system associated with the server node.
6. The method of claim 1, 2, 3, 4 or 5 further including,

5
10
15
20
25
30
35
40
45
50
55

436

upon detecting the received identification data, the server node (20) self-creating a communication line (175,180) to the peripheral device.

7. The method of claim 1, 2, 3, 4, 5 or 6 wherein the identification data is received (130) by the peripheral device (30) having broadcast (125) the identification data over the network (10). 5
8. The method of claim 1, 2, 3, 4, 5, 6 or 7 further including the server node (20) self-modifying (190,192,195) to embody changed configuration parameters of the peripheral device as identified from the received identification data. 10
9. An apparatus (50,55) for configuring a peripheral device (30) on a network (10) without user intervention, the network having a server node (20) connected thereto, the apparatus comprising: 15
 - (a) means (50,55) for enabling the server node (20) to receive identification data (130) for identifying the peripheral device (30) on the network (10); and, 20
 - (b) means (50,55) for enabling the server node (20) to self-install a driver (160) for the peripheral device (30) upon detecting the received identification data. 25
10. The apparatus of claim 9 further including means for enabling the server node (20) to self-modify (190,192,195) to embody changed configuration parameters of the peripheral device (30) as identified from the received identification data. 30

35

40

45

50

55

112

FIG. 1

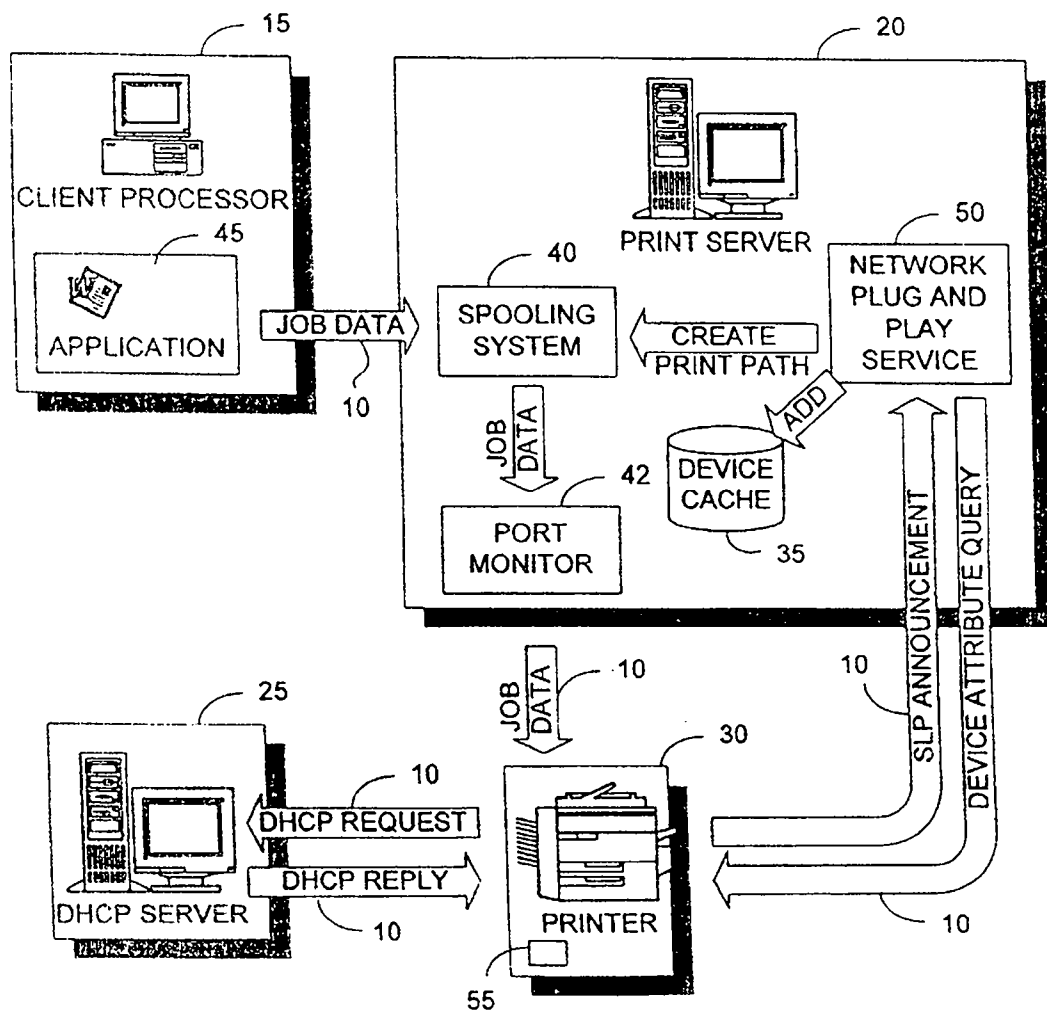


FIG. 2

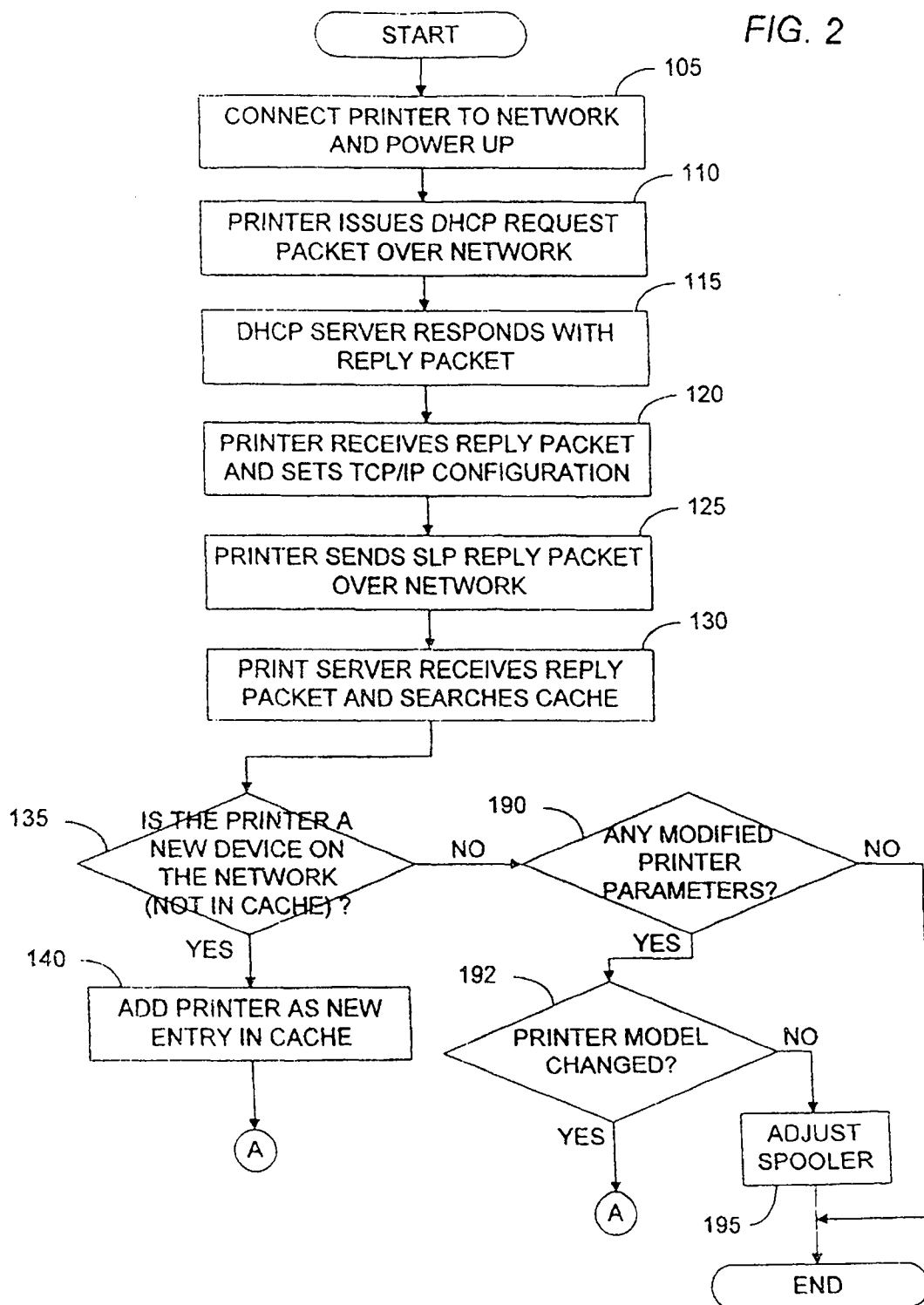
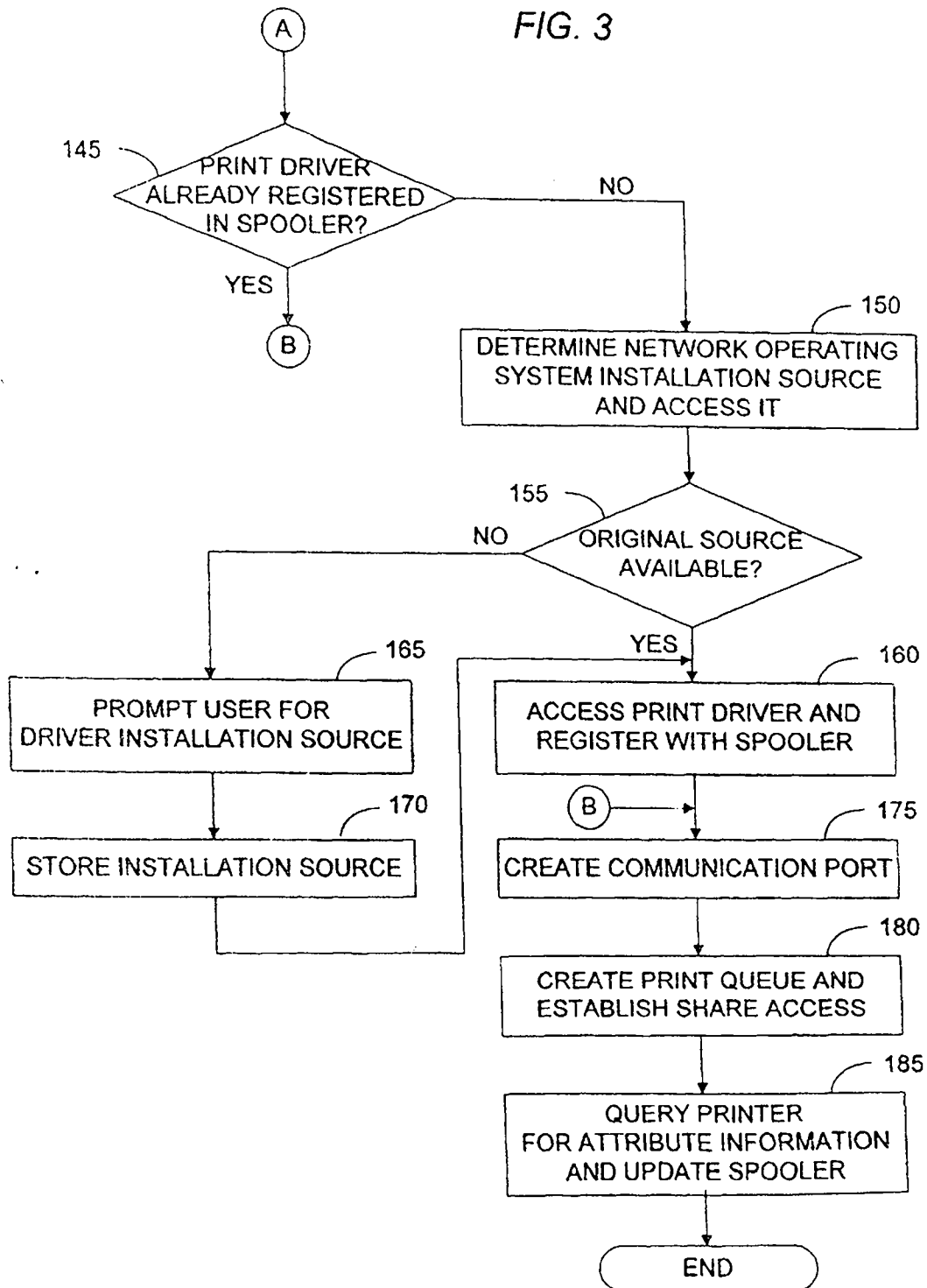


FIG. 3



440

EP 0 952 513 A1



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number:
EP 98 11 8576

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	US 5 655 148 A (RICHMAN DARRYL STEVEN ET AL) 5 August 1997 (1997-08-05) * figures 1-7 * * column 12. line 20 - column 34, line 41 *	1,3-6, 8-10	G06F3/12
A	US 5 564 109 A (SALES MILTON S ET AL) 8 October 1996 (1996-10-08) * figures 1,2 * * column 2. line 38 - column 3. line 64 *	1,2,5-7, 9,10	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G06F G06K
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 6 August 1999	Examiner Weiss, P
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure F : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EP 0 952 513 A1 (1999-08-05)

441

EP 0 952 513 A1

ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.

EP 98 11 8576

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

06-08-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5655148 A	05-08-1997	US 5787246 A	28-07-1998
		US 5748980 A	05-05-1998
		US 5793979 A	11-08-1998
		US 5819107 A	06-10-1998
		US 5809329 A	15-09-1998
US 5564109 A	08-10-1996	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

117

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30	A1	(11) International Publication Number: WO 00/51031 (43) International Publication Date: 31 August 2000 (31.08.00)		
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;"> (21) International Application Number: PCT/US00/04698 (22) International Filing Date: 25 February 2000 (25.02.00) (30) Priority Data: 09/258,242 26 February 1999 (26.02.99) US (63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application US Not furnished (CIP) Filed on Not furnished (71) Applicant (for all designated States except US): AMERICA ONLINE, INC. [US/US]; 22000 AOL Way, Duiles, VA 20166 (US). (72) Inventor; and (75) Inventor/Applicant (for US only): HENDREN, C., Hudson, III [US/US]; 1340 Old Grade Road, Strasburg, VA 22657 (US). (74) Agents: HAYDEN, John, F. et al.; Fish & Richardson, P.C., 601 Thirteenth Street N.W., Washington, DC 20005 (US). </td> <td style="width: 50%; border: none; vertical-align: top;"> (81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments. </td> </tr> </table>			(21) International Application Number: PCT/US00/04698 (22) International Filing Date: 25 February 2000 (25.02.00) (30) Priority Data: 09/258,242 26 February 1999 (26.02.99) US (63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application US Not furnished (CIP) Filed on Not furnished (71) Applicant (for all designated States except US): AMERICA ONLINE, INC. [US/US]; 22000 AOL Way, Duiles, VA 20166 (US). (72) Inventor; and (75) Inventor/Applicant (for US only): HENDREN, C., Hudson, III [US/US]; 1340 Old Grade Road, Strasburg, VA 22657 (US). (74) Agents: HAYDEN, John, F. et al.; Fish & Richardson, P.C., 601 Thirteenth Street N.W., Washington, DC 20005 (US).	(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
(21) International Application Number: PCT/US00/04698 (22) International Filing Date: 25 February 2000 (25.02.00) (30) Priority Data: 09/258,242 26 February 1999 (26.02.99) US (63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application US Not furnished (CIP) Filed on Not furnished (71) Applicant (for all designated States except US): AMERICA ONLINE, INC. [US/US]; 22000 AOL Way, Duiles, VA 20166 (US). (72) Inventor; and (75) Inventor/Applicant (for US only): HENDREN, C., Hudson, III [US/US]; 1340 Old Grade Road, Strasburg, VA 22657 (US). (74) Agents: HAYDEN, John, F. et al.; Fish & Richardson, P.C., 601 Thirteenth Street N.W., Washington, DC 20005 (US).	(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.			
(54) Title: PROXY SERVER AUGMENTING A CLIENT REQUEST WITH USER PROFILE DATA (57) Abstract A proxy server includes a database, a network interface, and a processor. The database includes records storing user profile information. The network interface is coupled to a network to exchange data with a client computer and with a target server. The processor is operatively coupled to the network interface, the database, and to a memory. The memory includes executable instructions for causing the processor to receive a data request from a client computer at the network interface, augment the data request by adding user profile information, and send the augmented data request to the network interface for delivery to the target server. A data transfer method performed at a proxy server includes intercepting a data request directed from a client computer to a target server. The intercepted data request is then augmented at the proxy server by adding user profile information and sent to a target server.				
<pre> graph TD 300[Establish User Profile Data 300] --> 301[HTTP Request Received 301] 301 --> 302[Identify User 302] 302 --> 303[Insert User Profile in Request 303] 303 --> 304[Forward Request To HTTP Server 304] </pre>				

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

PROXY SERVER AUGMENTING A CLIENT REQUEST WITH USER PROFILE DATA

BACKGROUND

Client computers can communicate with a server to remotely access information stored at the server. The transfer of information between the server and client computers may be provided using standard protocols and software applications. For example, a hypertext markup language (HTML) browser application at a client computer can communicate over the public Internet using TCP/IP and hypertext transfer protocols (HTTP) to receive web pages from a HTTP server. Web pages may include formatted text as well as multimedia elements, such as embedded graphics and sounds. Example browser applications include Netscape Navigator 4.0® and Microsoft Internet Explorer 4.0™.

When a server receives an information request from a client computer, the server may require information about a user of the client computer to respond to the information request. For example, a server providing a local news and weather service may need a user's home address in order to select appropriate news and weather data to send to the user. A server can obtain the needed user information by sending to the client computer a data input form that is displayed to the user. The input form may include a number of fields that the user can fill in with data and then send back to the server. The server may then use the data in the fields in responding to the requests.

Obtaining data using forms displayed to a user can be an unreliable, bothersome, and time consuming process. If a server requires a user to input lengthy personal data or requests data that was previously provided, the user may become frustrated with the information service system and discontinue or reduce use of the system. This may lead to reduced revenue for the information service system provider. Additionally, a user may provide intentionally or unintentionally incorrect information to a server thus decreasing a server's ability to reliably process information.

A server can use a web cookie to reduce repetitive data entry requests by that server. A web cookie is a token exchanged between a server and client computer that may be used to store data or to refer to and identify past transactions. A web cookie containing user information may be stored at a client computer by a server during a hypertext transfer

445

protocol (HTTP) transaction, and retrieved by the server during subsequent HTTP transactions. In general, web cookies are unique to particular servers and are used to store data related to a particular server-client pairing. A server may include user data in a web cookie and store that web cookie at a client computer. When the client sends a subsequent data request to the server, the server may access the web cookie that was previously stored at the client. However, a web server may be unable to properly interpret another web server's web cookie and, therefore, may still need to request data items that are found in other server's web cookies. Furthermore, since web cookies are stored at the client computer, they may be deleted or modified by a client computer user. Therefore, even though a server may store a web cookie at a client, the server cannot rely on a web cookie being present during subsequent transactions with the client.

The present inventors recognize that obtaining data about a user through forms and other direct input mechanisms can be time consuming and unreliable. Furthermore, the inventors recognize that existing method of storing user data and tracking user interactions, such as web cookie mechanisms can be insufficient. Consequently, improved means of providing information about a user are desired.

SUMMARY

In general, in one aspect, the invention features a data transfer method performed at a proxy server. The method includes intercepting a data request directed from a client computer to a target server. The intercepted data request is then augmenting at the proxy server by adding user profile information and sent to a target server.

Implementations may include one or more of the following features. User profile information may be added to the data request by adding a user profile field to the request. The user profile field may include a header identifying the field as a user profile field and the associated user profile data. The field may be a HTTP formatted field added to a HTTP request sent from a web browser to a server. In a secure implementation, the proxy server may encrypt the profile data to limit the target servers that can access the profile data. For example, an information service provider may implement a proxy server that adds encrypted profile data to a web request. The information server provider may then provide decryption information to a limited set of business partners allowed to access the profile data. To limit the servers that receive the profile data, the proxy server may maintain a list of web server

121

446

addresses and may only add the profile data to a request when the request is directed to a web server on the list.

Implementations also may include one or more of the following features. The proxy server may identify a relevant user profile to add to a data request based on a network address received along with request messages from a client. To do so, a unique network address may be associated with the user. The network address may be assigned to a client computer when the client computer establishes a data connection with a network access point. Login data may then be received from a user and sent to the proxy server along with the network address. The login data and network address may be stored in a proxy server database allowing the proxy server to associate particular login data items with particular client network addresses. This may allow the proxy server to determine a user associated with a particular data request.

In general, in another aspect, the invention features a computer program residing on a computer-readable medium. The program includes instructions for causing a computer to receive a data request directed to a target server from a client computer. The program instructions may then cause the data request to be augmented with user profile information and sent to the target server.

Implementations of the program may include one or more of the following features. The program may instruct the computer to receive login data and a unique network address and associate the login data and network address with a user of the client computer. The login data and network address may be stored in a database. The program may further cause the computer to access the login data by retrieving the record from the database based on receipt of the address assigned to the client computer in association with the receipt of the data request and to determine user profile information to be added to the data request based on the login data.

In general, in another aspect, the invention features a proxy server. The proxy server includes a database, a network interface, and a processor. The database includes records storing user profile information. The network interface is coupled to a network to exchange data with a client computer and with a target server. The processor is operatively coupled to the network interface, the database, and to a memory. The memory includes executable instructions for causing the processor to receive a data request from a client computer at the

172

network interface, augment the data request by adding user profile information, and send the augmented data request to the network interface for delivery to the target server. The memory may also include instructions for causing the processor to receive from the network interface user access data comprising a user identification and a network address associated with the user's connection to a network.

In general, in another aspect, the invention features a method of processing a data request. The method includes receiving a hypertext transfer protocol (HTTP) data request at a server from a proxy server, identifying a field added to the data request by the proxy server, determining a response to the data request based on the user profile data; and sending the response to the proxy server for delivery to a client computer. The added field includes user profile data and a header identifying the user profile data. The user profile data may be received in an encrypted form and decrypted by the server.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Implementations may provide advantages including reduction of redundant data entry, reduction of user data entry errors, secure storage of user data, and automated user profile determination and dissemination. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 shows computers connected by a network.
FIGS. 2A and 2B show exemplary HTTP request data.
FIGS. 3A and 3B are flowcharts.

DETAILED DESCRIPTION

Fig. 1 shows a network 100 that includes server computers 131-133 and client computers 111-113. Server computers 131-133 may execute hypertext transfer protocol (HTTP) server software to respond to data requests from HTTP-based web browsers executing at client computers 111-113. Client computers 111-113 can send HTTP data requests to servers 131-133 over data paths that include access connections 114-116, a service provider's point of presence (POP) 110, network 120, proxy server 117, and network 130. The service provider's POP 110 includes data communications equipment that enables

448

and regulates communication between client computers 111-113 and the service provider's network 120. For example, POP 110 may include dial-up modem banks, cable modem banks, wireless communications equipment, or other data transmission equipment.

Access to a service provider's POP 110 may be restricted to certain users of the client computers 111-113. To enforce access restrictions, POP 110 may implement security and authentication mechanisms such as login verification. A login verification mechanism may require a user to input a valid user name and password to obtain access to the service provider's network 120. If the user name and password are invalid, the user may be disconnected. Security and authentication mechanisms also may be implemented at a separate login server connected to the service provider's network 120 and/or POP 110.

After a valid connection has been established between a client computer and POP 110, data may be exchanged between software applications running on the client computer, and applications running on other computers on the service provider's network 120. The service provider's network 120 may be interconnected with another network 130 by a proxy server 117 that can exchange data between the service provider's network 120 and computers on another network 130.

Proxy server 117 can function as a surrogate for another computer. For example, proxy server 117 may intercept HTTP data requests sent between a browser application at client computer 111 and HTTP servers at computers 131-133. When the proxy server 117 intercepts a data request, it may attempt to fulfill the data request using data stored at a local database or hard disk drive 119. If the proxy server 117 has the needed data, the data can be returned to the client computer 111 without requiring further interaction with servers 131-133. If the proxy server 117 is unable to fulfill the request, it may forward the request to a server 131-133, receive a response from that server, and send the response to the client computer 111. The proxy server 117 also may store response data on a hard disk drive or local database 119 for future use. Networks 120 and 130 also may be interconnected by a gateway, bridge, router, or other interconnection device instead of, or in addition to, proxy server 117.

A server 131-133 may be configured to receive data requests from multiple client computers 111-113 which may be generated by multiple different users of those client computers. Access to particular server computers 131-133 may be restricted to particular

124

users of the client computers 111-113 similar to the manner in which access to POP 110 is restricted. Thus a user may provide name and password data when accessing the POP 110, may again provide name and password data when accessing server 131, and may yet again provide name and password data when accessing server 132. Implementations may also use
5 different access control mechanisms at POP 110 and at each server 131-133.

Servers 131-133 also may require other types of data from a user in order to process information requests. For example, a local weather information server may require a user to input the user's home address each time local weather is requested. A commercial web site may also require the home address in order to determine a user's shipping address. Input of
10 such user data may be time consuming and redundant.

User data entry may be reduced by storing user profile information at a proxy server 117 and automatically sending the profile information to servers when data request are made. Thus, an information service provider (ISP) may store user demographic data or other user data at proxy server 117 and automatically send that data to selected web sites along with
15 data request being sent to those web sites. In a HTTP implementation, data request may use the HTTP protocol. The proxy server 117 may intercept HTTP request sent from a web browser application at a client computer 111 to a HTTP server 131. The proxy 117 may then insert user profile information in the HTTP request and send the modified HTTP request to a server 131.

20 Fig. 2A shows a HTTP request 200. The HTTP request 200 includes a structured sequence of fields 201-203. Each field 201-203 includes a HTTP header and data associated with the header. HTTP headers provide a structured description of each HTTP request field and each field's data. For example, field 201 indicates that the HTTP request is a "GET" request to obtain a default web page from the server "www.acme-gizmos.com". Field 202
25 includes the HTTP header "User-Agent:" which indicating that the field's data "Mozilla/3.0 compatible" designates the type of browser generating the request 200. Similarly, field 203 includes the HTTP header "Cookie:" indicating that the data "been_here_before" is a web cookie.

The HTTP request 200 may be sent from a client computer 111 to proxy 117 and
30 forwarded through the proxy 117 for delivery to a server 131. The proxy server 117 can insert a user profile in the request 200 as the request is forwarded through the proxy.

450

Referring to Figs. 1 and 2A, when the proxy 117 receives the HTTP request 200, the proxy 117 can modify the request 200 to include user profile information. To modify the request 200, the proxy server 117 first determines a user associated with the request.

To determine the user associated with a HTTP request, a proxy server may use a table
5 or database that can associate user identity information with network connection information unique to each active user. The user identity information and network connection information may be provided to the proxy server 117 during user log-in to the POP 110 or to a login server.

For example, referring to Fig. 2A, when a user connects to a network service
10 provider's network by submitting name and password information to POP 110, a TCP/IP address and port, and/or other user-unique network connection information may be associated with the user's data transfer connection. The POP 110 or a login server may then send the user name and the user's unique network connection information to the proxy server 117 where it is stored in a database (step 300). When the proxy server 117 receives a
15 subsequent HTTP request 200 (step 301), the proxy server can identify the user associated with the request 200 by querying the database of stored name and network connection associations. The proxy server's database query may be based on network connection information associated with a TCP/IP data transfer connection over which the HTTP request was transferred.

When the proxy server 117 has identified the user (step 302), the proxy server then
20 may retrieve a user profile associated with that user from its database 119. The user profile information may be inserted by the proxy server as one or more fields in the HTTP request 200 (step 303). Fig. 2B shows a modified version 250 of the HTTP request 200 after user profile information has been inserted. The modified HTTP request 250 includes field 204
25 which includes user profile information inserted by the proxy server 117. Field 204 includes the HTTP header "User-Profile-Data:" indicating that the data "UserName=John_Doe, ZipCode=60609, ParentalControl=YoungTeen is user profile data. After the proxy server 117 has generated the modified request 250, it may then be forwarded to a server 131.

Referring to Fig. 2B, the modified request 250 is then received at a web server (step
30 321). Particular web servers on a network may or may not recognize the user profile data field 254. A web server that is not configured to recognize the user profile field 254 may

451

ignore the data in field 254 and instead process the other fields in the modified request 250. A web server that is configured to recognize the user profile field 254 can extract the profile data from the field (steps 322 and 324) and use the profile data to generate or customize data sent in response to a user (step 325). For example, a tourist information web server may

5 customize a page based on a user's age, and interest. Thus, for example, if a user's profile indicates that their interest include art and music, the tourist information web server may provide them with a list of museums and concert halls. On the other hand, if the user's interest indicates that they like canoeing and hiking, the tourist information server may provide them with a list of public parks. If the user profile field is not present in a web page,

10 the server may request user input to obtain needed data (step 322-323) and then generate a subsequent response to the user (step 325).

In a secure implementation, profile data sent from the proxy server 117 to a server 131-133 may be in an encrypted form. For example, field 206 may include a HTTP header indicating that the field contains user profile data in an encrypted form. Encryption of profile

15 data may help ensure that only authorized web servers 131-133 will be able to extract a user's profile data. Encryption may be performed using a public key cryptography algorithms. In a public key implementation, a web server 131-133 may publish its public key at a uniform resource locator (URL) address known to the proxy server. Other encryption algorithms may also be used. For example, the Rivest, Shamir, Adelman (RSA) encryption

20 technique may be used. Encryption of profile data helps to ensure that only authorized web servers can access the profile data. For example, in a commercial application, credit card information stored at the proxy server 117 may be encrypted by the proxy server prior to that information being sent to a web server in a user profile data field.

User profile data may be stored at the proxy server 117 by an information service

25 provider (ISP) through a database 119 access interface. The access interface may implement a query language such as structured query language (SQL) In such an implementation. SQL commands may be sent to the database 119 from a network configuration or database access terminal. Other method of provisioning profile data for use by the proxy server 117 may also be used. User profile data stored at the proxy server 117 may be updated by a computer

30 system coupled to the network 120, by the proxy server 117. or by a web server 131-133. The proxy server 117 may update stored user profile data by monitoring data requests from a

452

client and responses to those requests obtained from web servers. For example, the proxy server may update a list of visited web sites stored in a users' profile based on request received from a client computer. A web server 131-133 may also be configured to update profile information stored at the proxy server. For example, the web server, in responding to a data request, may insert a user data profile update field in the HTTP request data being sent to the proxy server 117. The proxy server may then extract the update field information and use it to update a stored profile. A web server may also update profile data by transmitting profile update information separately from data being sent to a user. For example, web server 131 may initiate a TCP/IP connection directly to a database or other application at the proxy 117 that can receive profile update information and update stored profile data at the proxy 117.

A proxy server 117 may selectively insert profile data in a HTTP request depending on the destination address of the request. For example, profile data may be inserted in a HTTP request being sent to server 131 but not in a request being sent to server 133. The proxy server 117 may determine web servers that are to receive profile data based on destination site filtering information stored at the proxy server. Thus, an on-line service provider (OSP) or information service provider (ISP) may establish filtering conditions that configure a proxy server 117 to send user profile data to business partners and affiliated web sites, but not to competitors. The filtering information may be a database of internet host names, internet protocol addresses, and/or wild-card characters to determine servers that are permitted and/or forbidden to receive profile data.

The profile data stored at the proxy server may include, for example, the login name and password of a user of the client computer, additional data maintained by an information service provider, such as demographic information, a history of the user's data requests, age, gender, interest, and information that is determined dynamically by the exchange of data between the proxy server and web servers.

The invention may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention may be performed by a programmable processor executing a program of instructions to

127

perform functions of the invention by operating on input data and generating output. The invention may advantageously be implemented in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer
5 program may be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language may be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions
10 and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing may
15 be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits).

A number of embodiments of the present invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, server, proxy, and client
20 protocols need not use the HTTP protocol. Alternate protocols and data formats may be used such as file transfer protocol (FTP) or network news transfer protocol (NNTP). Accordingly, other embodiments are within the scope of the following claims.

454

WHAT IS CLAIMED IS:

- 1 1. A data transfer method performed at a proxy server, the method comprising:
2 intercepting a data request from a client computer that is directed to a target server;
3 augmenting the data request by adding user profile information; and
4 sending the augmented data request to the target server.
- 1 2. The method of claim 1 further comprising:
2 establishing a data connection between the client computer and a network access point;
3 receiving login data from a client computer user; and
4 sending the login data to the proxy server.
- 1 3. The method of claim 2 further wherein:
2 establishing a data connection comprises assigning a network address to the client
3 computer;
4 receiving login data comprises receiving a user identification; and
5 sending the login data comprises sending the user identification and the network address.
- 1 4. The method of claim 3 wherein:
2 receiving a data request further comprises receiving the network address; and
3 augmenting the data request comprises retrieving the user profile information from a
4 proxy server database based on the received network address.
- 1 5. The method of claim 4 wherein augmenting based on the received network address
2 comprises retrieving the user identification from a proxy server database based on the
3 received network address and accessing user profile information based on the user
4 identification.
- 1 6. The method of claim 1 wherein the user profile information comprises data associated
2 with a current user of the client computer.

- 1 7. The method of claim 1 further comprising receiving the profile information at the proxy
2 server from another computer and storing the profile information in a proxy server
3 database.
- 1 8. The method of claim 1 wherein the data request comprises a plurality of fields each
2 comprising a header and data associated with the header.
- 1 9. The method of claim 8 wherein the data request is a hypertext transfer protocol (HTTP)
2 data request, the target server is a HTTP server, the client computer comprises a web
3 browser application, and each header in the plurality of fields comprises a HTTP data
4 request header.
- 1 10. The method of claim 8 wherein adding the user profile information comprises adding a
2 field. the field comprising a profile header identifying the added field as a user profile
3 field and profile data associated with the user.
- 1 11. The method of claim 10 wherein the profile data comprises encrypted data.
- 1 12. The method of claim 11 further comprising:
2 receiving the augmented data request at the target server; and
3 decrypting the encrypted profile data.
- 1 13. The method of claim 1 further comprising identifying the target server as a server
2 permitted to receive user profile information.
- 1 14. A computer program residing on a computer-readable medium, comprising instructions
2 for causing a computer to:
3 receive a data request from a client computer directed at a target server;
4 augment the data request by adding user profile information; and

- 5 send the augmented data request to the target server.
- 1 15. The computer program of 14 further comprising instructions for causing a computer to:
2 receive a network address assigned to the client computer and login data associated with
3 a user of the client computer; and
4 store the network address and login data in a record in a database.
- 1 16. The computer program of 14 further comprising instructions for causing the computer to
2 access the login data by retrieving the record from the database based on receipt of the
3 address assigned to the client computer in association with the receipt of the data request.
- 1 17. The computer program of claim 16 further comprising instructions for causing the
2 computer to determine user profile information to be added to the data request based on
3 the login data.
- 1 18. A proxy server comprising:
2 a database comprising records storing user profile information;
3 a network interface operatively coupled to a network to exchange data with a client
4 computer and with a target server; and
5 a processor operatively coupled to the network interface, the database, and a memory
6 comprising executable instructions for causing the processor to receive a data request
7 from a client computer at the network interface, augment the data request by adding
8 user profile information, and send the augmented data request to the network
9 interface for delivery to the target server.
- 1 19. The server of claim 18 wherein the memory further comprises instructions for causing the
2 processor to receive from the network interface user access data comprising a user
3 identification and a network address associated with the user's connection to a network.
- 1 20. A method of processing a data request comprising:

2 receiving a hypertext transfer protocol (HTTP) data request at a server from a proxy
 3 server;
 4 identifying a field added to the data request by the proxy server and comprising user
 5 profile data and a header identifying the user profile data
 6 determining a response to the data request based on the user profile data; and
 7 sending the response to the proxy server for delivery to a client computer.

1 21. The method of claim 20 wherein the user profile data is encrypted and determining a
 2 response based on the user profile data comprises decrypting the user profile data.



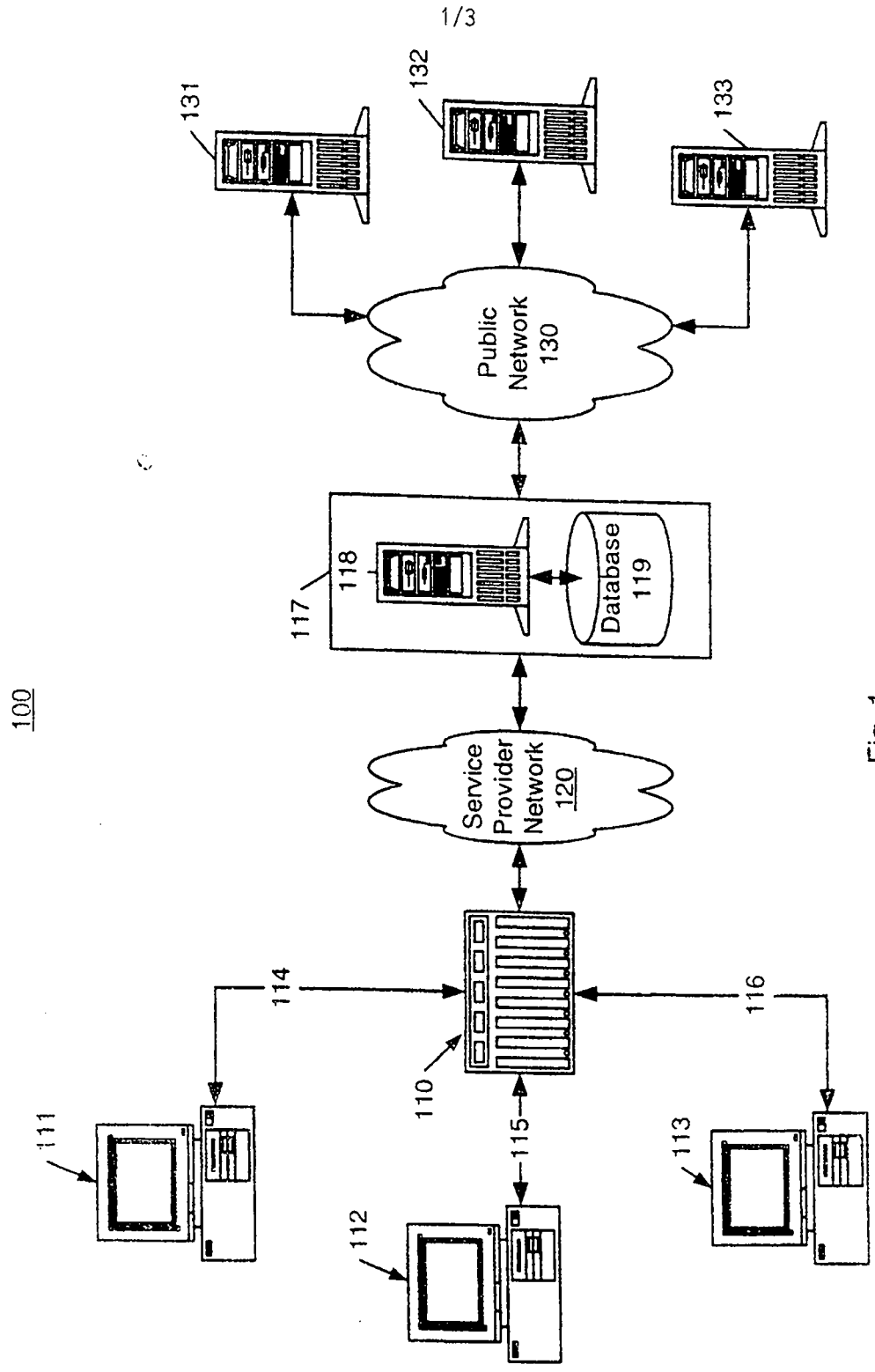


Fig. 1

459

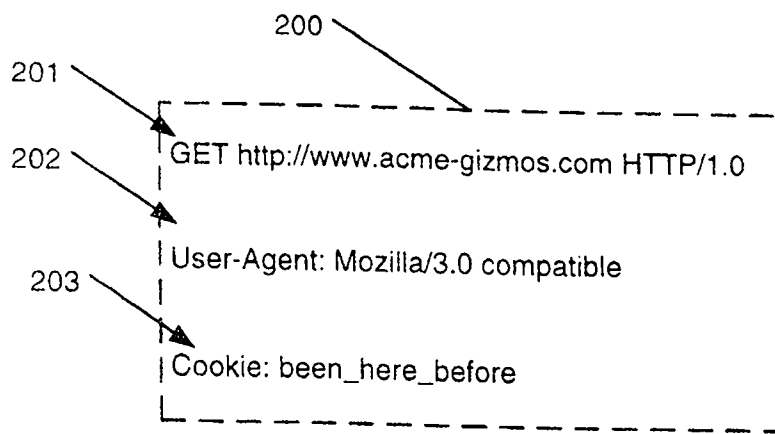


Fig. 2A

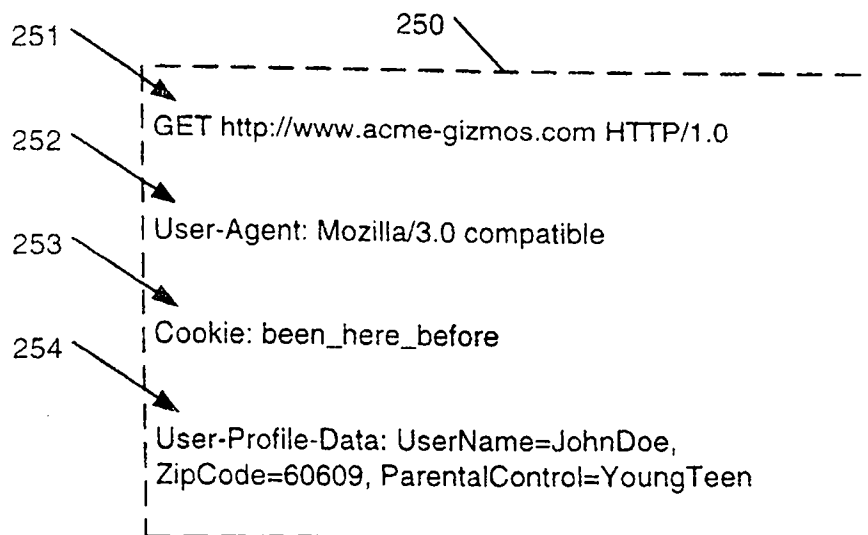


Fig. 2B

135

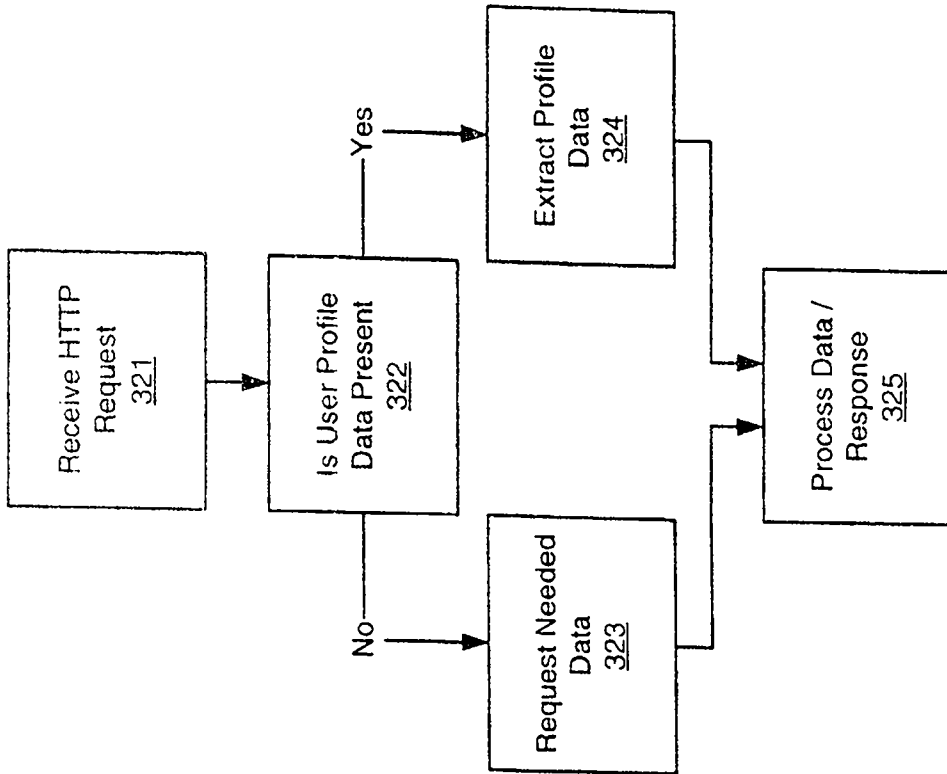


Fig. 3B

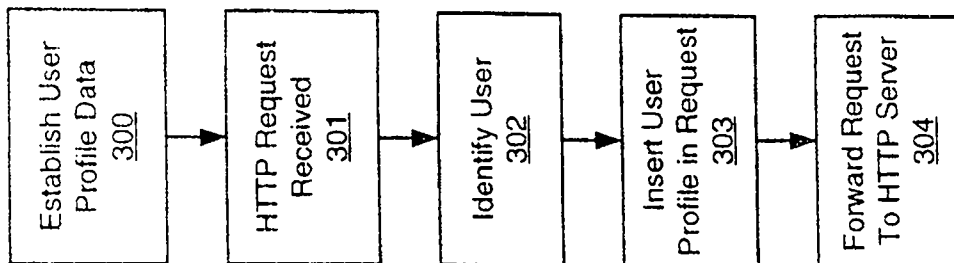


Fig. 3A

460

U36

INTERNATIONAL SEARCH REPORT

Int. l. Application No

PCT/US 00/04698

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

WPI Data, EPO-Internal, PAJ, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PETITCOLAS F A P ET AL: "WebGroup: a secure group access control tool for the World-Wide Web" PROCEEDINGS SEVENTH IEEE INTERNATIONAL WORKSHOP ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES (WET ICE '98) (CAT. NO.98TB100253), PROCEEDINGS OF WET ICE'98 - IEEE SEVENTH INTERNATIONAL WORKSHOP ON ENABLING TECHNOLOGIES: INFRAS, pages 301-305, XP002142486, 1998, Los Alamitos, CA, USA, IEEE Comput. Soc, USA ISBN: 0-8186-8751-7	1,2, 6-10, 13-15, 18-20
A	page 302, left-hand column, line 14 -page 303, right-hand column, line 9; figures 1-6 ---	11,12,21
	---	-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

13 July 2000

Date of mailing of the international search report

26/07/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5616 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Polzer, A

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/04698

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 586 260 A (HU WEI-MING) 17 December 1996 (1996-12-17) abstract column 2, line 1 - line 49 column 3, line 58 -column 6, line 45; figures 1-4	1,2,6,7, 14,18
X	GABBER E ET AL: "How to make personalized Web browsing simple, secure and anonymous" FINANCIAL CRYPTOGRAPHY FIRST INTERNATIONAL CONFERENCE, FC '97. PROCEEDINGS, FINANCIAL CRYPTOGRAPHY FIRST INTERNATIONAL CONFERENCE, FC'97. PROCEEDINGS, ANGUILLA, 24-28 FEB. 1997, 'Online! pages 17-31, XP002059819 1997, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-63594-7 Retrieved from the Internet: <URL:http://www.math.tau.ac.il/{matias/rec ent.html}> 'retrieved on 2000-07-13! page 20, line 32 -page 25, line 7; figures 1-4	1,2,6,8, 9,14,18
A	WO 98 33130 A (MOTOROLA INC) 30 July 1998 (1998-07-30) abstract page 8, line 29 -page 13, line 12; figures 2,3 page 14, line 11 -page 15, line 34; figure 5 page 18, line 32 -page 25, line 8; figures 7-9	1-7, 13-19

463

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/04698

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5586260	A	17-12-1996	NONE	
WO 9833130	A	30-07-1998	US 6049821 A	11-04-2000
			AU 714951 B	13-01-2000
			AU 5801098 A	18-08-1998
			EP 1010099 A	21-06-2000

139

1

**METHOD, APPARATUS, AND PRODUCT
FOR LEASING OF DELEGATION
CERTIFICATES IN A DISTRIBUTED
SYSTEM**

RELATED APPLICATIONS

This is a continuation-in-part of U.S. patent application Ser. No. 08/729,421, filed on Oct. 11, 1996, U.S. Pat. No. 5,832,529, which is incorporated herein by reference.

The following identified U.S. patent applications are relied upon and are incorporated by reference in this application.

Provisional U.S. Patent Application No. 60/076,048, entitled "Distributed Computing System," filed on Feb. 26, 1998.

U.S. patent application Ser. No. 09/044,923, entitled "Method and System for Leasing Storage," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,834, entitled "Method, Apparatus and Product for Leasing of Group Membership in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,916, entitled "Methods and Systems For Distributed Failure Detection and Recovery Using Leasing," U.S. Pat. No. 6,016,500, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,933, entitled "Method for Transporting Behavior in Event Based System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,919, entitled "Deferred Reconstruction of Objects and Remote Loading for Event Notification in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,938, entitled "Methods and Apparatus for Remote Method Invocation," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/045,652, entitled "Method and System for Deterministic Hashes to Identify Remote Methods," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,790, entitled "Method and Apparatus for Determining Status of Remote Objects in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,917, entitled "Suspension and Continuation of Remote Methods," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching in a Database," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database," abandoned, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,945, entitled "Method and System for Typesafe Attribute Matching in a Database," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,931, entitled "Dynamic Lookup Service in a Distributed System," pending, and filed on the same date herewith.

2

U.S. patent application Ser. No. 09/044,939, entitled "Apparatus and Method for Providing Downloadable Code for Use in Communicating with a Device in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,826, entitled "Method and System for Facilitating Access to a Lookup Service," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,932, entitled "Apparatus and Method for Dynamically Verifying Information in a Distributed System," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/030,840, entitled "Method and Apparatus for Dynamic Distributed Computing Over a Network," and filed on Feb. 26, 1998, pending.

U.S. patent application Ser. No. 09/044,936, entitled "An Interactive Design Tool for Persistent Shared Memory Spaces," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,934, entitled "Polymorphic Token-Based Control," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,915, entitled "Stack-Based Access Control," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,944, entitled "Stack-Based Security Requirements," pending, and filed on the same date herewith.

U.S. patent application Ser. No. 09/044,837, entitled "Per-Method Designation of Security Requirements," pending, and filed on the same date herewith.

BACKGROUND OF THE INVENTION

A. Field of the Invention

This invention generally relates to data processing systems and, more particularly, to leasing delegation certificates in data processing systems.

B. Description of the Related Art

Proper resource management is an important aspect to efficient and effective use of computers. In general, resource management involves allocating resources (e.g., memory) in response to requests as well as deallocating resources at appropriate times, for example, when the requesters no longer require the resources. In general, the resources contain data referenced by computational entities (e.g., applications, programs, applets, etc.) executing in the computers.

In practice, when applications executing on computers seek to refer to resources, the computers must first allocate or designate resources so that the applications can properly refer to them. When the applications no longer refer to a resource, the computers can deallocate or reclaim the resource for reuse. In computers each resource has a unique "handle" by which the resource can be referenced. The handle may be implemented in various ways, such as an address, array index, unique value, pointer, etc.

Resource management is relatively simple for a single computer because the events indicating when resources can be reclaimed, such as when applications no longer refer to them or after a power failure, are easy to determine. Resource management for distributed systems connecting multiple computers is more difficult because applications in several different computers may be using the same resource.

Disconnects in distributed systems can lead to the improper and premature reclamation of resources or to the failure to reclaim resources. For example, multiple applica-

3

tions operating on different computers in a distributed system may refer to resources located on other machines. If connections between the computers on which resources are located and the applications referring to those resources are interrupted, then the computers may reclaim the resources prematurely. Alternatively, the computers may maintain the resources in perpetuity, despite the extended period of time that applications failed to access the resources.

These difficulties have led to the development of systems to manage network resources, one of which is known as "distributed garbage collection." That term describes a facility provided by a language or runtime system for distributed systems that automatically manages resources used by an application or group of applications running on different computers in a network.

In general, garbage collection uses the notion that resources can be freed for future use when they are no longer referenced by any part of an application. Distributed garbage collection extends this notion to the realm of distributed computing, reclaiming resources when no application on any computer refers to them.

Distributed garbage collection must maintain integrity between allocated resources and the references to those resources. In other words, the system must not be permitted to deallocate or free a resource when an application running on any computer in the network continues to refer to that resource. This reference-to-resource binding, referred to as "referential integrity," does not guarantee that the reference will always grant access to the resource to which it refers. For example, network failures can make such access impossible. The integrity, however, guarantees that if the reference can be used to gain access to any resource, it will be the same resource to which the reference was first given.

Distributed systems using garbage collection must also reclaim resources no longer being referenced at some time in the finite future. In other words, the system must provide a guarantee against "memory leaks." A memory leak can occur when all applications drop references to a resource, but the system fails to reclaim the resource for reuse because, for example, of an incorrect determination that some application still refers to the resource.

Referential integrity failures and memory leaks often result from disconnections between applications referencing the resources and the garbage collection system managing the allocation and deallocation of those resources. For example, a disconnection in a network connection between an application referring to a resource and a garbage collection system managing that resource may prevent the garbage collection system from determining whether and when to reclaim the resource. Alternatively, the garbage collection system might mistakenly determine that, since an application has not accessed a resource within a predetermined time, it may collect that resource. A number of techniques have been used to improve the distributed garbage collection mechanism by attempting to ensure that such mechanisms maintain referential integrity without memory leaks. One conventional approach uses a form of reference counting, in which a count is maintained of the number of applications referring to each resource. When a resource's count goes to zero, the garbage collection system may reclaim the resource. Such a reference counting scheme only works, however, if the resource is created with a corresponding reference counter. The garbage collection system in this case increments the resource's reference count as additional applications refer to the resource, and decrements the count when an application no longer refers to the resource.

4

Reference counting schemes, however, especially encounter problems in the face of failures that can occur in distributed systems. Such failures can take the form of a computer or application failure or network failure that prevent the delivery of messages notifying the garbage collection system that a resource is no longer being referenced. If messages go undelivered because of a network disconnect, the garbage collection system does not know when to reclaim the resource.

To prevent such failures, some conventional reference counting schemes include "keep-alive" messages, which are also referred to as "ping back." According to this scheme, applications in the network send messages to the garbage collection system overseeing resources and indicate that the applications can still communicate. These messages prevent the garbage collection system from dropping references to resources. Failure to receive such a "keep-alive" message indicates that the garbage collection system can decrement the reference count for a resource and, thus, when the count reaches zero, the garbage collection system may reclaim the resource. This, however, can still result in the premature reclamation of resources following reference counts reaching zero from a failure to receive "keep-alive" messages because of network failures. This violates the referential integrity requirement.

Another proposed method for resolving referential integrity problems in garbage collection systems is to maintain not only a reference count but also an identifier corresponding to each computational entity referring to a resource. See A. Birrell, et al., "Distributed Garbage Collection for Network Objects," No. 116, digital Systems Research Center, Dec. 15, 1993. This method suffers from the same problems as the reference counting schemes. Further, this method requires the addition of unique identifiers for each computational entity referring to each resource, adding overhead that would unnecessarily increase communication within distributed systems and add storage requirements (i.e., the list of identifiers corresponding to applications referring to each resource).

SUMMARY OF THE INVENTION

In accordance with the present invention, referential integrity is guaranteed without costly memory leaks by leasing resources for a period of time during which the parties in a distributed system, for example, an application holding a reference to a resource and the garbage collection system managing that resource, agree that the resource and a reference to that resource will be guaranteed. At the end of the lease period, the guarantee that the reference to the resource will continue lapses, allowing the garbage collection system to reclaim the resource. Because the application holding the reference to the resource and the garbage collection system managing the resource agree to a finite guaranteed lease period, both can know when the lease and, therefore, the guarantee, expires. This guarantees referential integrity for the duration of a reference lease and avoids the concern of failing to free the resource because of network errors. In addition to memory, the leasing technique can be applied to delegation certificates.

Consistent with an alternative embodiment of the present invention, as embodied and broadly described herein, a method for leasing delegation certificates is provided. This method comprises the steps of receiving a lease request from a client specifying a resource and a lease period, determining a lease period during which the client has authority to request from a server access to the resource, advising the

5

client of the granted lease period, granting the client a delegation certificate which the client can use to access the resource from the server.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

FIG. 1 is a flow diagram of the steps performed by the application call processor according to an implementation of the present invention;

FIG. 2 is a flow diagram of the steps performed by the server call processor to process dirty calls according to the implementation of the present invention;

FIG. 3 is a flow diagram of the steps performed by the server call processor to process clean calls according to the implementation of the present invention;

FIG. 4 is a flow diagram of the steps performed by the server call processor to initiate a garbage collection process according to the implementation of the present invention;

FIG. 5 is a diagram of a preferred flow of calls within a distributed processing system;

FIG. 6 is a block diagram of the components of the implementation of a method invocation service according to the present invention;

FIG. 7 is a diagram of a distributed processing system that can be used in an implementation of the present invention; and

FIG. 8 is a diagram of the individual software components in the platforms of the distributed processing system according to the implementation of the present invention.

FIG. 9 is a diagram of a data processing system for leasing delegation certificates in a distributed processing system that can be used in an alternative embodiment of the present invention; and

FIG. 10 is a flow diagram of the steps performed by the delegator process when another process (potential delegatee) requests a lease according to an alternative embodiment of the present invention; and

FIG. 11A and FIG. 11B represent a flow diagram of the steps performed by a process (potential delegatee) when requesting a lease from the delegator process according to an alternative embodiment of the present invention; and

FIG. 12 is a flow diagram of the steps performed by the server when a delegatee requests access to a resource according to an alternative embodiment of the present invention.

DETAILED DESCRIPTION

Reference will now be made in detail to an implementation of the present invention as illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

The present invention may be implemented by computers organized in a conventional distributed processing system architecture. The architecture for and procedures to implement this invention, however, are not conventional, because they provide a distributed garbage collection scheme that ensures referential integrity and eliminates memory leaks.

A. Overview

A method invocation (MI) component located in each of the computers in the distributed processing system imple-

6

ments the distributed garbage collection scheme of this invention. The MI component may consist of a number of software modules preferably written in the JAVA™ programming language.

In general, whenever an application in the distributed processing system obtains a reference to a distributed resource, by a name lookup, as a return value to some other call, or another method, and seeks to access the resource, the application makes a call to the resource or to an MI component managing the resource. That MI component, called a managing MI component, keeps track of the number of outstanding references to the resource. When the number of references to a resource is zero, the managing MI component can reclaim the resource. The count of the number of references to a resource is generally called the "reference count" and the call that increments the reference count may be referred to as a "dirty call."

When an application no longer requires a distributed resource, it sends a different call to the resource or the managing MI component. Upon receipt of this call, the managing MI component decrements the reference count for the resource. This call to drop a reference may be referred to as a "clean call."

In accordance with an implementation of the present invention, a dirty call can include a requested time interval, called a lease period, for the reference to the resource. Upon receipt of the dirty call, the managing MI component sends a return call indicating a period for which the lease was granted. The managing MI component thus tracks the lease period for those references as well as the number of outstanding references. Consequently, when the reference count for a resource goes to zero or when the lease period for the resource expires, the managing MI component can reclaim the resource.

B. Procedure

An application call processor in an MI component performs the steps of the application call procedure 100 illustrated in FIG. 1. The server call processor in the managing MI component performs the steps of the procedures 200, 300, and 400 illustrated in FIGS. 2-4, respectively. The managing MI component's garbage collector performs conventional procedures to reclaim resources previously bound to references in accordance with instructions from the server call processor. Accordingly, the conventional procedures of the garbage collector will not be explained.

1. Application Call Processor

FIG. 1 is a flow diagram of the procedure 100 that the application call processor of the MI component uses to handle application requests for references to resources managed by the same or another MI component located in the distributed processing system.

After an application has obtained a reference to a resource, the application call processor sends a dirty call, including the resource's reference and a requested lease period to the managing MI component for the resource (step 110). The dirty call may be directed to the resource itself or to the managing MI component.

The application call processor then waits for and receives a return call from the managing MI component (step 120). The return call includes a granted lease period during which the managing MI component guarantees that the reference of the dirty call will be bound to its resource. In other words, the managing MI component agrees not to collect the resource corresponding to the reference of a dirty call for the grant period. If the managing MI component does not provide a grant period, or rejects the request for a lease, then

the application call processor will have to send another dirty call until it receives a grant period.

The application call processor monitors the application's use of the reference and, either when the application explicitly informs the application call processor that the reference is no longer required or when the application call processor makes this determination on its own (step 130), the application call processor sends a clean call to the managing MI component (step 140). In a manner similar to the method used for dirty calls, the clean call may be directed to the referenced resource and the managing MI component will process the clean call. Subsequently, the application call processor eliminates the reference from a list of references being used by the application (step 150).

If the application is not yet done with the reference (step 130), but the application call processor determines that the grant period for the reference is about to expire (step 160), then the application call processor repeats steps 110 and 120 to ensure that the reference to the resource is maintained by the managing MI component on behalf of the application.

2. Server Call Processor

The MI component's server call processor performs three main procedures: (1) handling dirty calls; (2) handling incoming clean calls; and (3) initiating a garbage collection cycle to reclaim resources at the appropriate time.

(i) Dirty Calls

FIG. 2 is a flow diagram of the procedure 200 that the MI component's server call processor uses to handle requests to reference resources, i.e., dirty calls, that the MI software component manages. These requests come from application call processors of MI components in the distributed processing system, including the application call processor of the same MI component as the server call processor handling requests.

First, the server call processor receives a dirty call (step 210). The server call processor then determines an acceptable grant period (step 220). The grant period may be the same as the requested lease period or some other time period. The server call processor determines the appropriate grant period based on a number of conditions including the amount of resource required and the number of other grant periods previously granted for the same resource.

When the server call processor determines that a resource has not yet been allocated for the reference of a dirty call (step 230), the server call processor allocates the required resource (step 240).

The server call processor then increments a reference count corresponding to the reference of a dirty call (step 250), sets the acceptable grant period for the reference-to-resource binding (step 260), and sends a return call to an application call processor with the grant period (step 270). In this way, the server call processor controls incoming dirty calls regarding references to resources under its control.

Applications can extend leases by sending dirty calls with an extension request before current leases expire. As shown in procedure 200, a request to extend a lease is treated just like an initial request for a lease. An extension simply means that the resource will not be reclaimed for some additional interval of time, unless the reference count goes to zero.

(ii) Clean Calls

The MI component's server call processor also handles incoming clean calls from application call processors. When an application in the distributed processing system no longer requires a reference to a resource, it informs the MI component managing the resource for that reference so that the resource may be reclaimed for reuse. FIG. 3 is a flow diagram of the procedure 300 with the steps that the MI component's server call processor uses to handle clean calls.

When the server call processor receives a clean call with a reference to a resource that the MI component manages (step 310), the server call processor decrements a corresponding reference count (step 320). The clean call may be sent to the resource, with the server call processor monitoring the resource and executing the procedure 300 to process the call. Subsequently, the server call processor sends a return call to the MI component that sent the clean call to acknowledge receipt (step 330). In accordance with this implementation of the present invention, a clean call to drop a reference may not be refused, but it must be acknowledged.

(iii) Garbage Collection

The server call processor also initiates a garbage collection cycle to reclaim resources for which it determines that either no more references are being made to the resource or that the agreed lease period for the resource has expired. The procedure 400 shown in FIG. 4 includes a flow diagram of the steps that the server call processor uses to initiate a garbage collection cycle.

The server call processor monitors reference counts and granted lease periods and determines whether a reference count is zero for a resource managed by the MI component, or the grant period for a reference has expired (step 410). When either condition exists, the server call processor initiates garbage collection (step 420) of that resource. Otherwise, the server call processor continues monitoring the reference counts and granted lease periods.

C. Call Flow

FIG. 5 is a diagram illustrating the flow of calls among MI components within the distributed processing system. Managing MI component 525 manages the resources 530 by monitoring the references to those resources 530 (see garbage collect 505). Because the managing MI components 525 manages the resources, the server call processor of managing MI component 525 performs the operations of this call flow description.

FIG. 5 also shows that applications 510 and 540 have corresponding MI components 515 and 545, respectively. Each of the applications 510 and 540 obtains a reference to one of the resources 530 and seeks to obtain access to one of the resources 530 such that a reference is bound to the corresponding resource. To obtain access, applications 510 and 540 invoke their corresponding MI components 515 and 545, respectively, to send dirty calls 551 and 571, respectively, to the MI component 525. Because the MI components 515 and 525 handle application requests for access to resources 530 managed by another MI component, such as managing MI component 525, the application call processors of MI components 515 and 545 perform the operations of this call flow description.

In response to the dirty calls 551 and 571, managing MI component 525 sends return calls 552 and 572, respectively, to each of the MI components 515 and 545, respectively. The dirty calls include granted lease periods for the references of the dirty calls 551 and 571.

Similarly, FIG. 5 also shows MI components 515 and 545 sending clean calls 561 and 581, respectively, to managing MI component 525. Clean calls 561 and 581 inform managing MI component 525 that applications 510 and 540, respectively, no longer require access to the resource specified in the clean calls 561 and 581. Managing MI component 525 responds to clean calls 561 and 581 with return calls 562 and 582, respectively. Return calls 562 and 582 differ from return calls 552 and 572 in that return calls 562 and 582 are simply acknowledgments from MI component 525 of the received clean calls 561 and 581.

Both applications 510 and 540 may request access to the same resource. For example, application 510 may request access to "RESOURCE(1)" while application 540 was previously granted access to that resource. MI component 525 handles this situation by making the resource available to both applications 510 and 540 for agreed lease periods. Thus, MI component 525 will not initiate a garbage collection cycle to reclaim the "RESOURCE(1)" until either applications 510 and 540 have both dropped their references to that resource or the latest agreed periods has expired, whichever event occurs first.

By permitting more than one application to access the same resource simultaneously, the present invention also permits an application to access a resource after it sent a clean call to the managing MI component dropping the reference to the resource. This occurs because the resource is still referenced by another application or the reference's lease has not yet expired so the managing MI component 525 has not yet reclaimed the resource. The resource, however, will be reclaimed after a finite period, either when no more applications have leases or when the last lease expires.

D. MI Components

FIG. 6 is a block diagram of the modules of an MI component 600 according to an implementation of the present invention. MI component 600 can include a reference component 605 for each reference monitored, application call processor 640, server call processor 650, and garbage collector 660.

Reference component 605 preferably constitutes a table or comparable structure with reference data portions 610, reference count 620, and grant period register 630. MI component 600 uses the reference count 620 and grant period 630 for each reference specified in a corresponding reference data portion 610 to determine when to initiate garbage collector 660 to reclaim the corresponding resource.

Application call processor 640 is the software module that performs the steps of procedure 100 in FIG. 1. Server call processor 650 is the software module that performs the steps of procedures 200, 300, and 400 in FIGS. 2-4. Garbage collector 660 is the software module that reclaims resources in response to instructions from the server call processor 650, as explained above.

E. Distributed Processing System

FIG. 7 illustrates a distributed processing system 50 which can be used to implement the present invention. In FIG. 7, distributed processing system 50 contains three independent and heterogeneous platforms 100, 200, and 300 connected in a network configuration represented by the network cloud 55. The composition and protocol of the network configuration represented in FIG. 7 by the cloud 55 is not important as long as it allows for communication of the information between platforms 700, 800 and 900. In addition, the use of just three platforms is merely for illustration and does not limit the present invention to the use of a particular number of platforms. Further, the specific network architecture is not crucial to this invention. For example, another network architecture that could be used in accordance with this invention would employ one platform as a network controller to which all the other platforms would be connected.

In the implementation of distributed processing system 50, platforms 700, 800 and 900 each include a processor 710, 810, and 910 respectively, and a memory, 750, 850, and 950, respectively. Included within each processor 710, 810, and 910, are applications 720, 820, and 920, respectively, operating systems 740, 840, and 940, respectively, and MI components 730, 830, and 930, respectively.

Applications 720, 820, and 920 can be programs that are either previously written and modified to work with the present invention, or that are specially written to take advantage of the services offered by the present invention. Applications 720, 820, and 920 invoke operations to be performed in accordance with this invention.

MI components 730, 830, and 930 correspond to the MI component 600 discussed above with reference to FIG. 6.

Operating systems 740, 840, and 940 are standard operating systems tied to the corresponding processors 710, 810, and 910, respectively. The platforms 700, 800, and 900 can be heterogeneous. For example, platform 700 has an UltraSparc microprocessor manufactured by Sun Microsystems Corp. as processor 710 and uses a Solaris operating system 740. Platform 800 has a MIPS microprocessor manufactured by Silicon Graphics Corp. as processor 810 and uses a Unix operating system 840. Finally, platform 900 has a Pentium microprocessor manufactured by Intel Corp. as processor 910 and uses a Microsoft Windows 95 operating system 940. The present invention is not so limited and could accommodate homogenous platforms as well.

Sun, Sun Microsystems, Solaris, Java, and the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UltraSparc and all other SPARC trademarks are used under license and are trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Memories 750, 850, and 950 serve several functions, such as general storage for the associated platform. Another function is to store applications 720, 820, and 920. MI components 730, 830, and 930, and operating systems 740, 840, and 940 before execution by the respective processor 710, 810, and 910. In addition, portions of memories 750, 850, and 950 may constitute shared memory available to all of the platforms 700, 800, and 900 in network 50.

E. MI Services

The present invention may be implemented using a client-server model. The client generates requests, such as the dirty calls and clean calls, and the server responds to requests.

Each of the MI components 730, 830 and 930 shown in FIG. 7 preferably includes both client components and server components. FIG. 8, which is a block diagram of a client platform 1000 and a server platform 1100, applies to any two of the platforms 700, 800, and 900 in FIG. 7.

Platforms 1000 and 1100 contain memories 1050 and 1150, respectively, and processors 1010 and 1110, respectively. The elements in the platforms 1000 and 1100 function in the same manner as similar elements described above with reference to FIG. 7. In this example, processor 1010 executes a client application 1020 and processor 1110 executes a server application 1120. Processors 1010 and 1110 also execute operating systems 1040 and 1140, respectively, and MI components 1030 and 1130, respectively.

MI components 1030 and 1130 each include a server call processor 1031 and 1131, respectively, an application call processor 1032 and 1132, respectively, and a garbage collector 1033 and 1133, respectively. Each of the MI components 1030 and 1130 also contains reference components, including reference data portions 1034 and 1134, respectively, reference counts 1035 and 1135, respectively, and grant period registers 1036 and 1136, respectively, for each reference that the respective MI component 1030 or 1130 monitors.

Application call processors 1032 and 1132 represent the client service and communicate with server call processors

1031 and 1131, respectively, which represent the server service. Because platforms 1000 and 1100 contain a server call processor, an application call processor, a garbage collector, and reference components, either platform can act as a client or a server.

For purposes of the discussion that follows, however, platform 1000 is designated the client platform and platform 1100 is designated as the server platform. In this example, client application 1020 obtains references to distributed resources and uses MI component 1030 to send dirty calls to the resources managed by MI component 1130 of server platform 1100.

Additionally, server platform 1100 may be executing a server application 1120. Server application 1120 may also use MI component 1130 to send dirty calls, which may be handled by MI component 1130 when the resources of those dirty calls are managed by MI component 1130. Alternatively, server application 1120 may use MI component 1130 to send dirty calls to resources managed by MI component 1030.

Accordingly, server call processor 1031, garbage collector 1033, and reference count 1035 for MI component 1030 of client platform 1000 are not active and are therefore presented in FIG. 8 as shaded. Likewise, application call processor 1132 of MI component 1130 of the server platform 1100 is shaded because it is also dormant.

When client application 1020 obtains a reference corresponding to a resource, application call processor 1032 sends a dirty call, which server call processor 1131 receives. The dirty call includes a requested lease period. Server call processor 1131 increments the reference count 1135 for the reference in the dirty call and determines a grant period. In response, server call processor 1131 sends a return call to application call processor 1030 with the grant period. Application call processor 1032 uses the grant period to update recorded grant period 1035, and to determine when the resource corresponding to the reference of its dirty call may be reclaimed.

Server call processor 1131 also monitors the reference counts and grant periods corresponding to references for resources that it manages. When one of its reference counts 1135 is zero, or when the grant period 1135 for a reference has expired, whichever event occurs first, server call processor 1131 may initiate the garbage collector 1133 to reclaim the resource corresponding to the reference that has a reference count of zero or an expired grant period.

The leased-reference scheme according to the implementation of the present invention does not require that the clocks on the platforms 1000 and 1100 involved in the protocol be synchronized. The scheme merely requires that they have comparable periods of increase. Leases do not expire at a particular time, but rather expire after a specific time interval. As long as there is approximate agreement on the interval, platforms 1000 and 1100 will have approximate agreement on the granted lease period. Further, since the timing for the lease is, in computer terms, fairly long, minor differences in clock rate will have little or no effect.

The transmission time of the dirty call can affect the protocol. If MI component 1030 holds a lease to reference and waits until just before the lease expires to request a renewal, the lease may expire before the MI component 1130 receives the request. If so, MI component 1130 may reclaim the resource before receiving the renewal request. Thus, when sending dirty calls, the sender should add a time factor to the requested lease period in consideration of transmission time to the platform handling the resource of a dirty call so that renewal dirty calls may be made before the lease period for the resource expires.

F. Conclusion

In accordance with the present invention a distributed garbage collection scheme ensures referential integrity and eliminates memory leaks by providing granted lease periods corresponding to references to resources in the distributed processing system such that when the granted lease periods expire, so do the references to the resources. The resources may then be collected. Resources may also be collected when they are no longer being referenced by processes in the distributed processing system with reference to counters assigned to the references for the resources.

Alternative Embodiment of the Present Invention

The leasing technique, described above, relates to garbage collection. However, an alternative embodiment of the present invention, as described below, can be used with delegation certificates.

A delegation certificate allows one actor ("a delegator") with sufficient privilege to access a resource to delegate its authority to access this resource to another actor ("a delegatee") who then accesses the resource on behalf of the delegator.

However, for security purposes, the delegator may not want to delegate to the delegatee carte blanche permission to access the resource for fear the delegatee may abuse its privilege either intentionally or unintentionally. Thus, the delegator may want to impose limits on the delegatee's access, such as the type of access permitted or the length of time access is permitted. The leasing of delegation certificates allows the delegator to control and limit the delegatee's access, thus providing additional security.

Delegation certificates can be leased to access various resources, such as files. An example of a delegation follows: a delegator may have confidential tax files managed by a file system manager. By prior negotiation, the file system manager will only grant access to these files to the owner, the delegator. However, the owner may need the taxes to be calculated by a tax program, so the owner delegates authority to the tax program, the delegatee, to access the tax files for a limited time, until April 15. Accordingly, the owner grants the tax program permission, via a delegation certificate, to access the files controlled by the file system manager until April 15th. This is accomplished by leasing the delegation certificate to the tax program such that the lease expires on April 15th. If the tax program attempts to access the file after this date, the lease expires and it is prevented from doing so by the file system manager.

The leasing of delegation certificates allows the owner to control or limit access to the files by the tax program. More specifically, the tax program requests a lease from the owner for access to the files stored with the file system manager for read access until April 15th. If a lease is granted, the owner sends to the tax program a delegation certificate that indicates the tax program is entitled to read-only access to the owner's files.

The file system manager has the responsibility of authenticating the delegation certificate as well as to determine the type and length of the tax program's access. At no time can the tax program directly change the contents of the delegation certificate. However, the tax program can request the owner to renew the lease (i.e., if the tax program requires additional time to access the file) or to cancel the lease (i.e., if the tax program's access is completed before the lease period expires).

The delegation certificate is an object that proves the identity of the delegator and has a set of corresponding rights

469

154

13

attached to it. In particular, the delegation certificate object contains a principal identifier, some means for proving the identification, and a specification of the rights. The specification of the rights includes methods for determining the type of access requested and the duration of a lease. In addition, the object includes methods for canceling a lease and for renewing a lease. Finally, the methods may generate exceptions that, when accessed, set forth the reason why invocation of the method was unsuccessful.

The object is an instance of a class that may be extended in many ways to offer more functionality, but the basic class is defined as follows:

```

interface Lease {
    obj FileHandle;
    public void getIdentifer ();
    public void getAuthentication ();
    public void getAccessType ();
    public long getDuration ();
    public void renew (long renewDuration) throws
        LeaseDeniedException,
        UnknownLeaseException,
        RemoteException;
    public void cancel () throws
        UnknownLeaseException,
        RemoteException;
}

```

The principal identifier gives the delegatee the appearance of being the delegator when the delegatee communicates with the system manager. The integrity of the identification is assured by any number of known authentication methods, such as public-key, challenge-response protocol, or shared secret technology.

Invoking the access type method provides the type of access the delegator permits. This method can be invoked by whoever has the delegation certificate, either the delegator, the delegatee, or the file system manager. For instance, the delegates will invoke the method to determine what type of access it is permitted to seek from the file system manager. The file system manager will invoke the method to determine what type of access it is permitted to grant. For example, the delegator may deem a particular file as read-only access. In this case, the file system manager will only allow read access for a subsequently granted lease for that particular file. Conversely, an attempt by the delegatee to write to that storage location would not be permitted by the file system manager.

Invoking the duration method provides the length of the granted lease period. This period represents the most recent lease granted by the delegator.

The renew method permits the renew of the lease, asking for more time, without having to re-initiate the original lease request. Situations where the delegatee may desire to renew the lease include when the original lease proves to be insufficient (i.e., the delegatee requires additional use of the storage location), or when only a partial lease (i.e., less than the requested lease) is granted.

In addition, the renew method can be continually invoked in order to obtain sequential lease periods. The renew method, however, cannot be invoked if the delegatee does not have an active lease. Also, the renew method has no return value; if the renewal is granted, the new lease period will be reflected in the lease object on which the call was made. If the delegator is unable or unwilling to renew the lease, the reason is set forth in the LeaseDeniedException generated by the renew method.

14

The cancel method is invoked when there is still time left on the lease, but the delegatee no longer desires access. The cancel method may also be invoked by the delegator if, for instance, it wants to cancel the delegatee's access. Thus, cancel allows the file, for example, to be reclaimed. In contrast, upon the end of a lease, (i.e., natural termination occurs), there is no notification obligation by the delegatee.

FIG. 9 depicts a data processing system 9000 suitable for use for by an alternative embodiment of the present invention. The data processing system 9000 includes a computer system 9002 connected to the Internet 9004. The computer system 9002 includes a memory 9010, a secondary storage device 9018, a central processing unit (CPU) 9024, an input device 9026, and a video display 9022. The secondary storage device 9018 further includes a number of files 9020. The memory 9010 further includes a delegator program 9008, a delegatee program 9010, and an operating system 9014 containing a file system manager 9016. The file system manager 9016 manages files 9020 on the secondary storage device 9018. The delegates 9010 requests access to a secondary storage device 9018 by requesting a lease from the delegator 9008. In response, the delegator 9008 may either grant or deny the lease as further described below. If the delegator 9008 grants the lease to the delegatee 9010, the delegatee 9010 then requests access to the secondary storage device 9018 from the file system manager 9016. One skilled in the art will appreciate that computer 9000 may contain additional or different components.

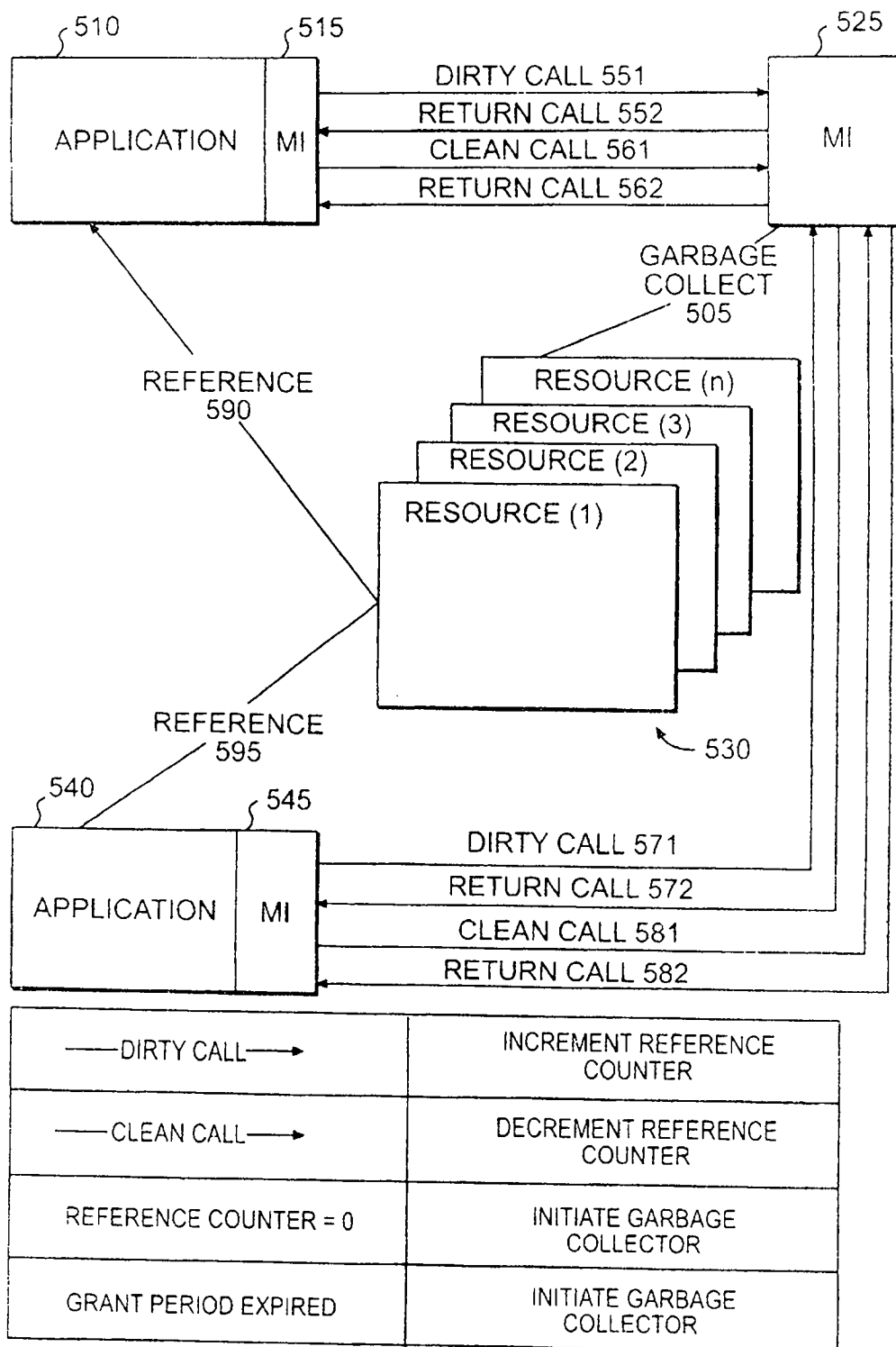
Although aspects of the alternative embodiment are described as being stored in memory 9010, one skilled in the art will appreciate that these aspects may also be stored in other computer readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet 9004; or other forms of RAM or ROM.

FIG. 10 depicts a flow chart of the steps performed by the delegator 9008 when receiving a lease request from the delegatee 9010. The first step performed by the delegator is to request the delegatee to access a resource, such as a file, on the delegator's behalf (step 10002). At some point later, the delegator will receive a lease request from the delegates (step 10004). This request is a function call that includes a number of parameters including (1) the requested file the delegatee wishes to lease, (2) the desired lease period, and (3) the type of access the delegates desires.

The requested file parameter contains an indication of the file to be leased. The desired lease period contains an amount of time the delegatee wants to utilize the file. The type of access requested indicates the type of access the client requested. For example, the delegatee may request read access or write access. To form a valid request, the delegatee request must contain both the file desired and the desired lease. After receiving the request, the delegator examines the parameters to verify the propriety of the request (step 10006).

After examining the parameters, the delegator determines if the request is, in fact, proper (step 10008). For example, the delegator checks if the requested file is a file that the delegator has the ability to lease. Also, the delegator verifies that some lease period is specified. Additionally, the delegator checks if the type of access requested is available. If the delegator determines that the lease request is improper, the delegator generates an exception (step 10010) and processing ends.

If the delegator determines that the lease request is proper, the delegator determines a sufficient lease period (step

**FIG. 5**

472

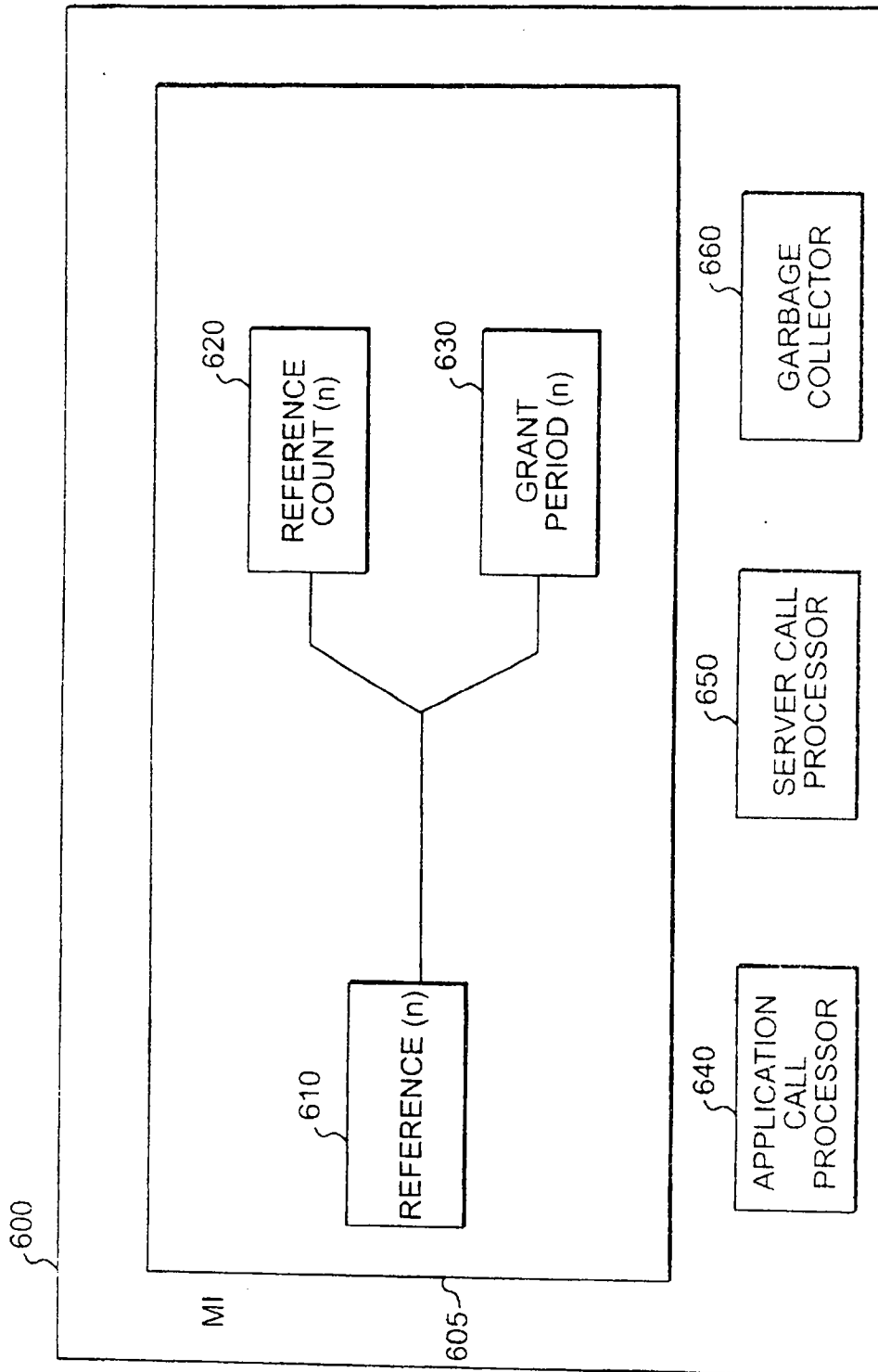


FIG. 6

473

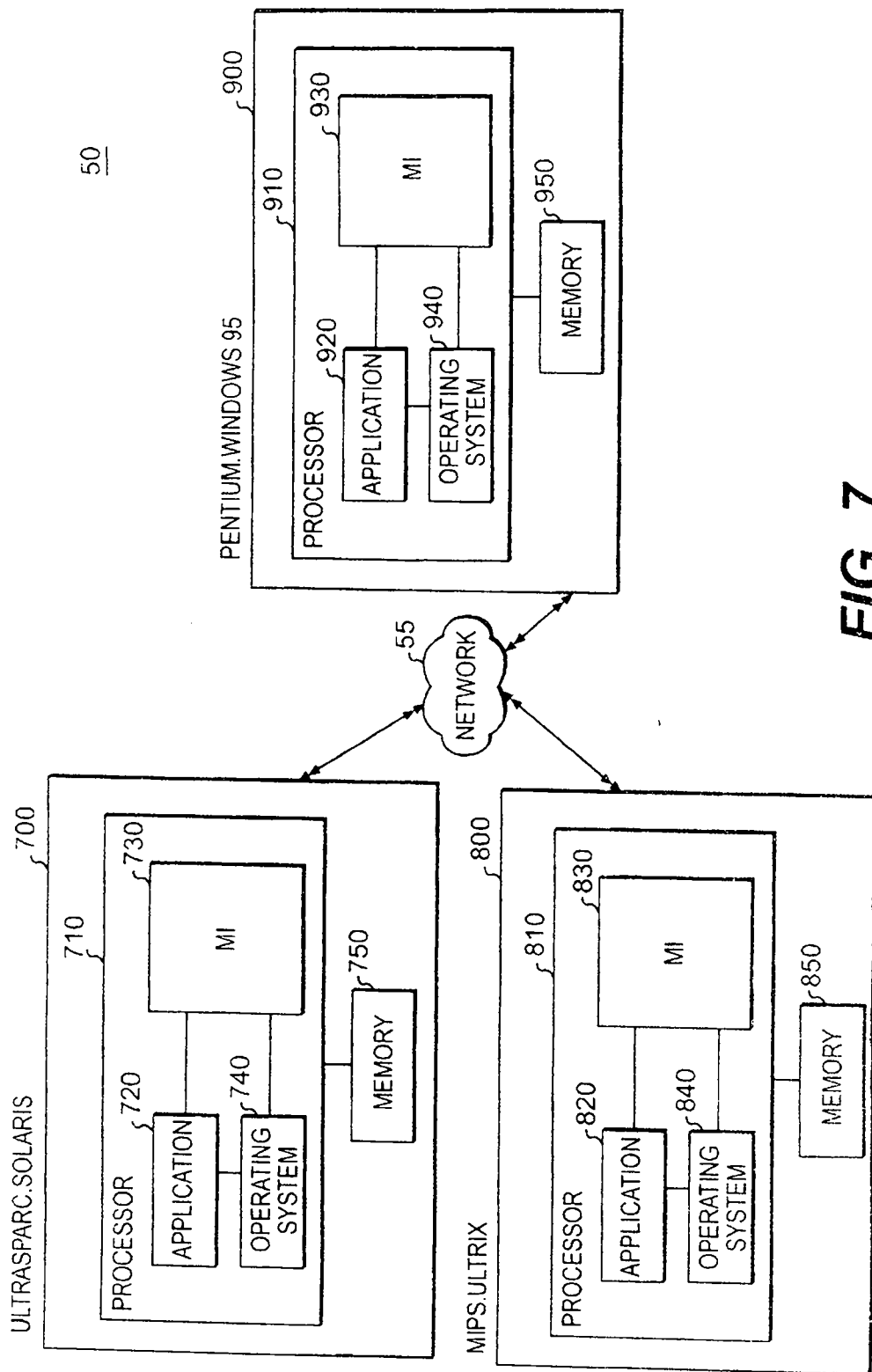


FIG. 7

474

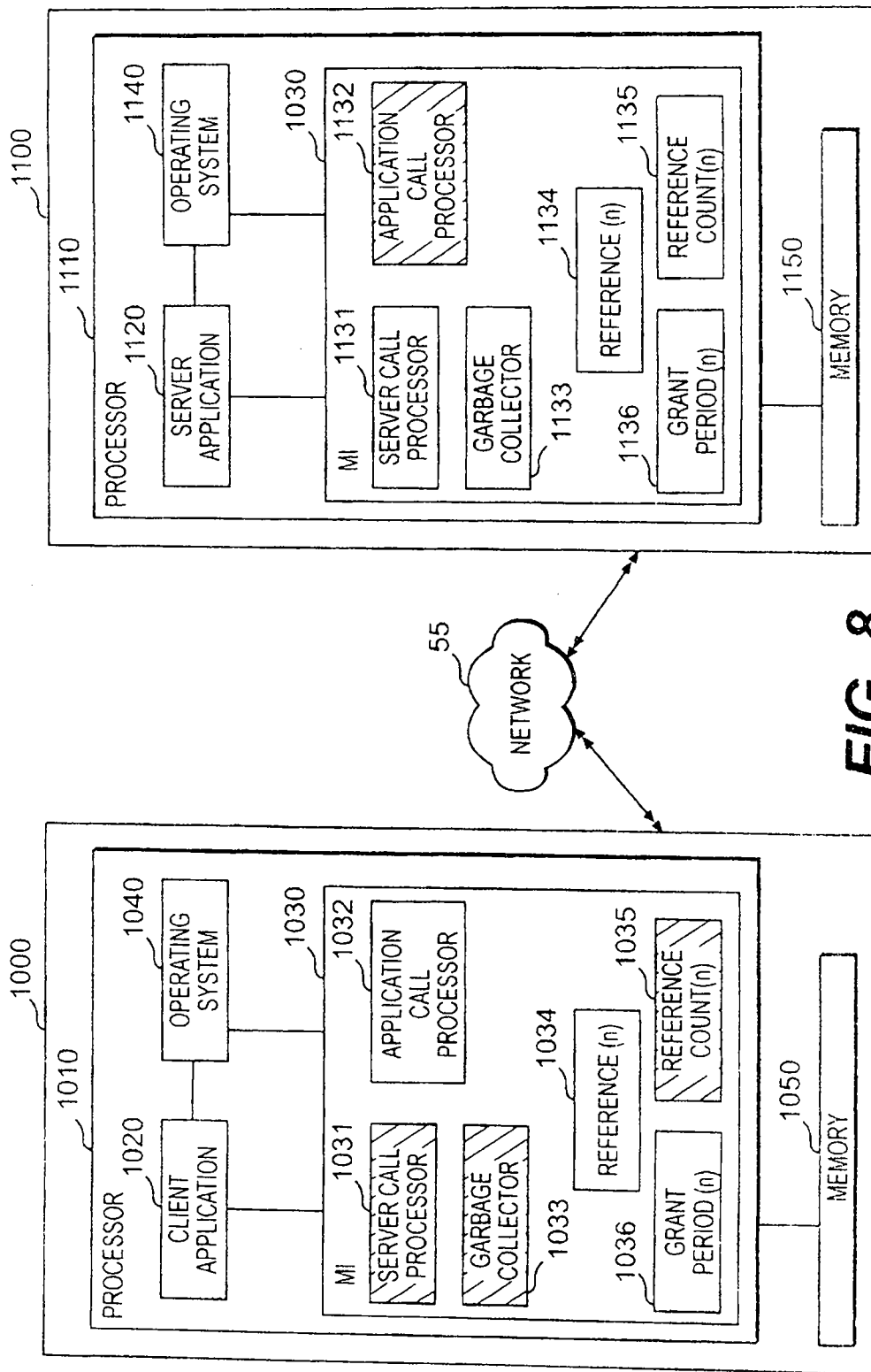


FIG. 8

475

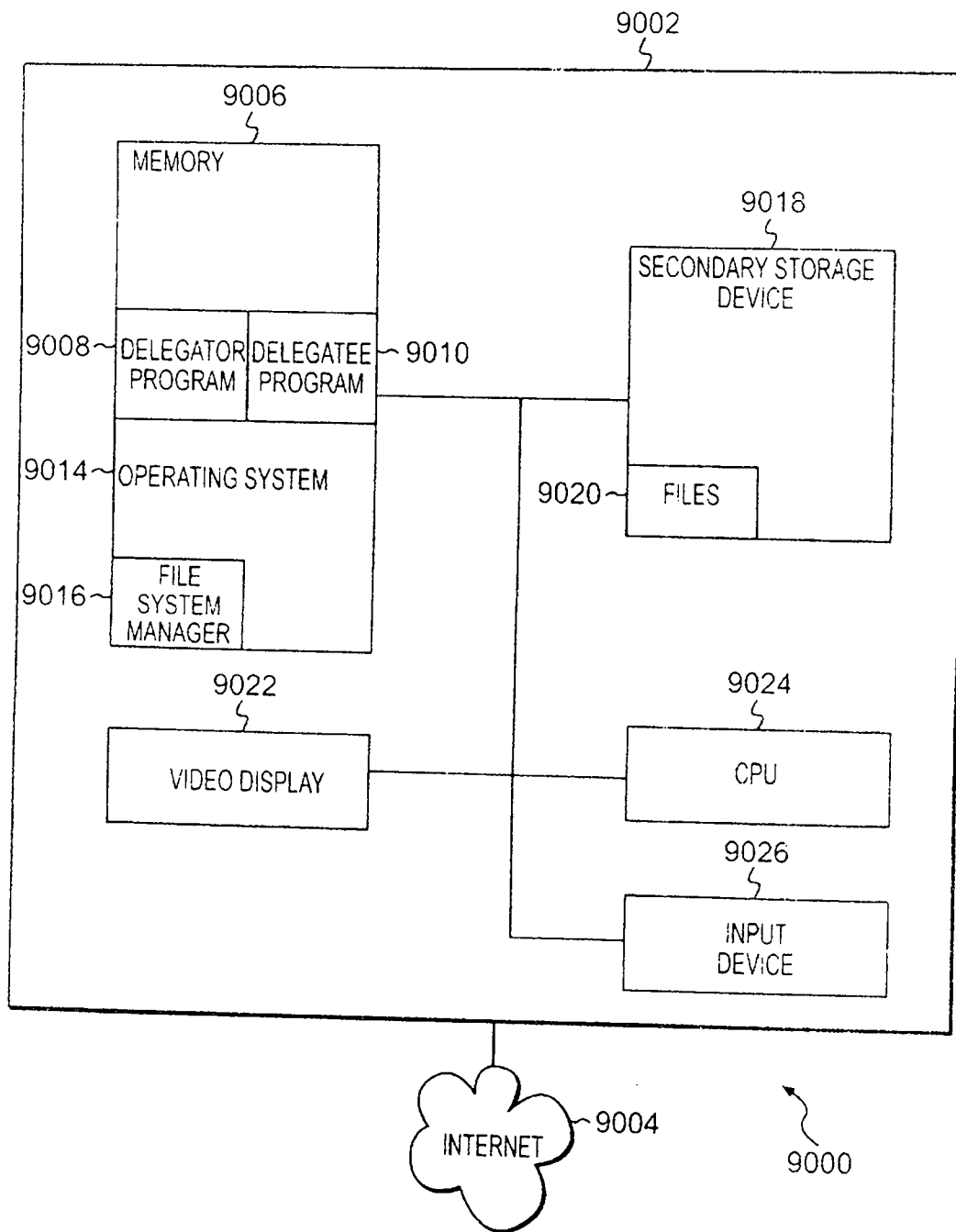
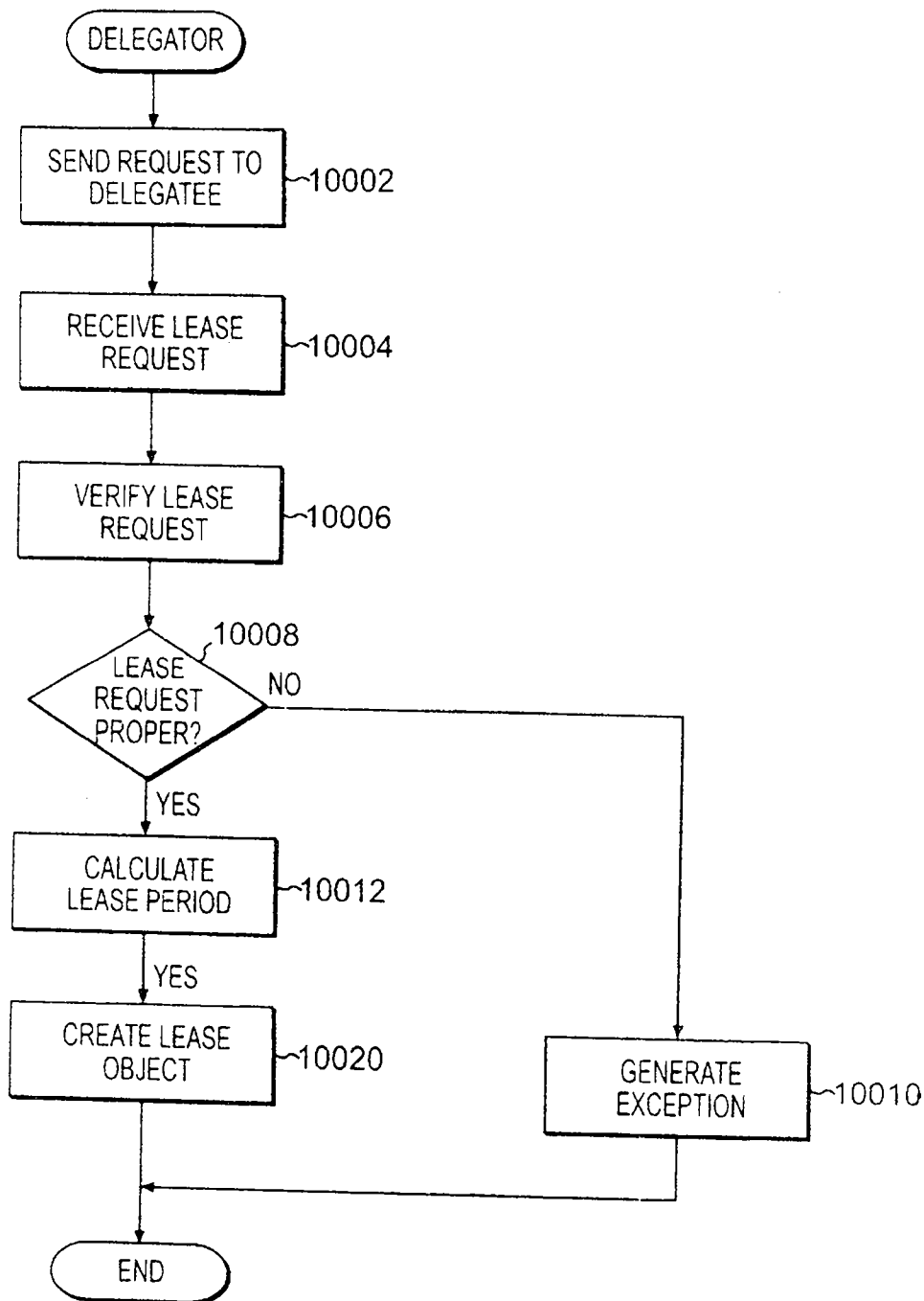
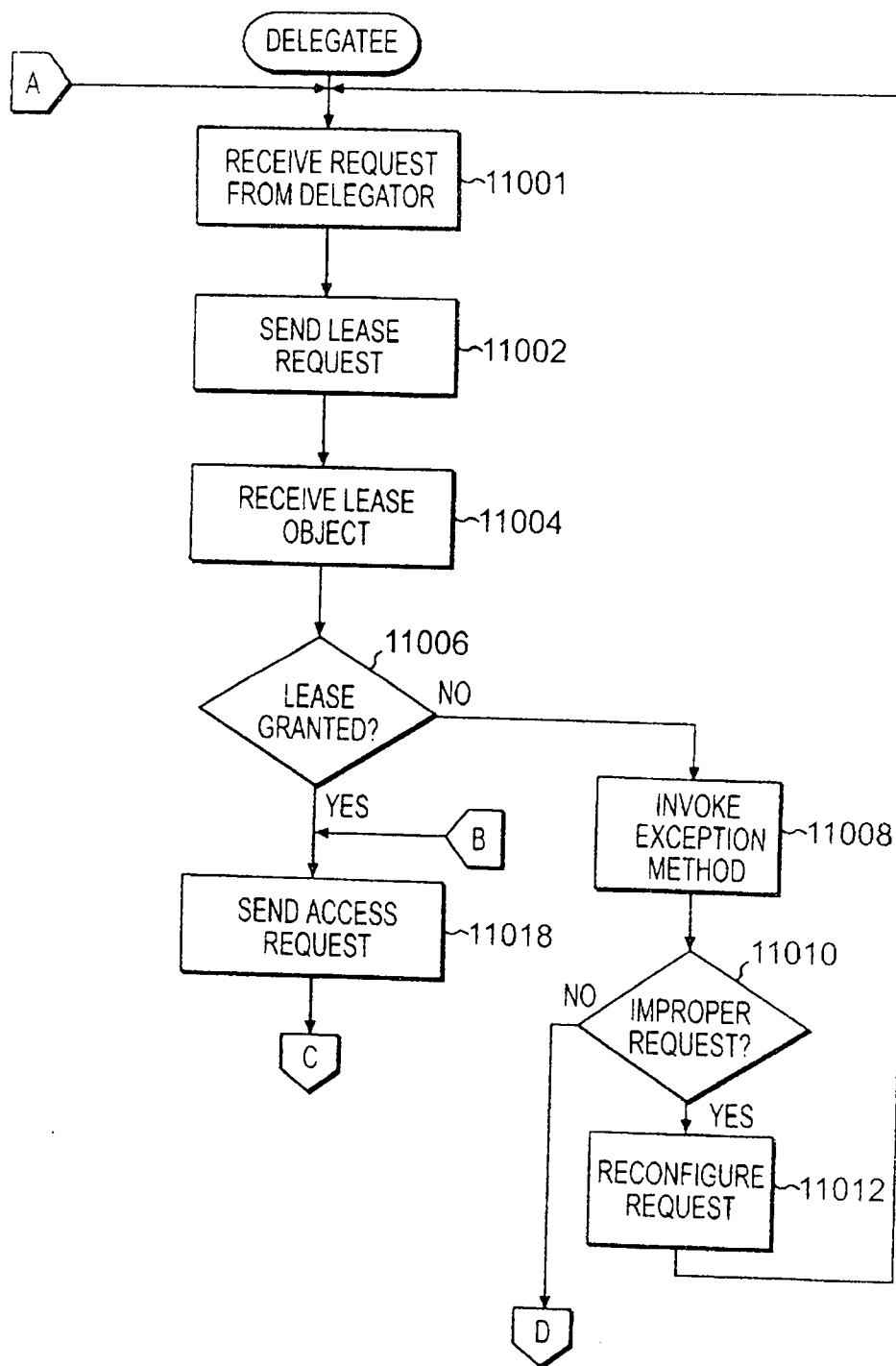
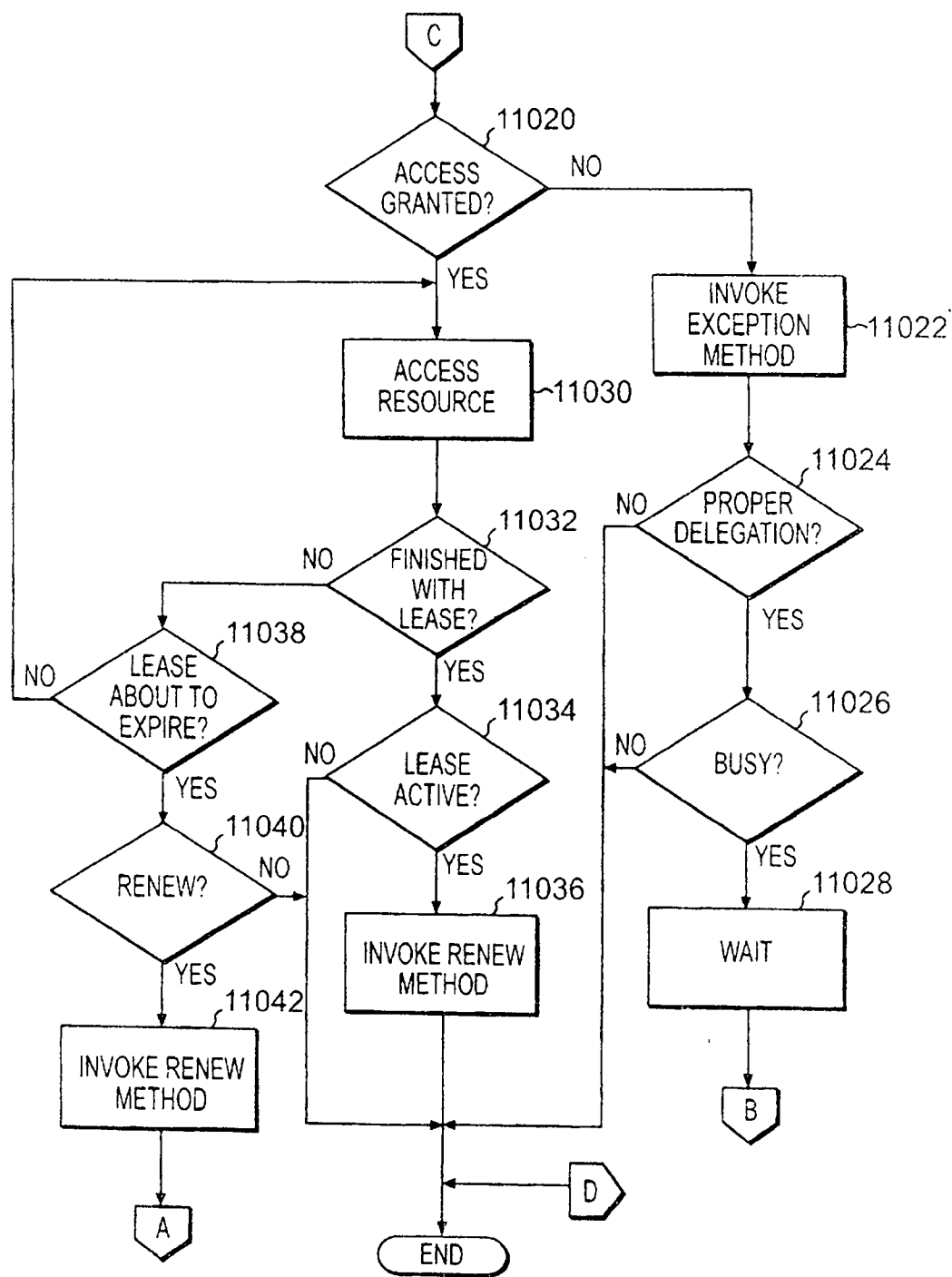
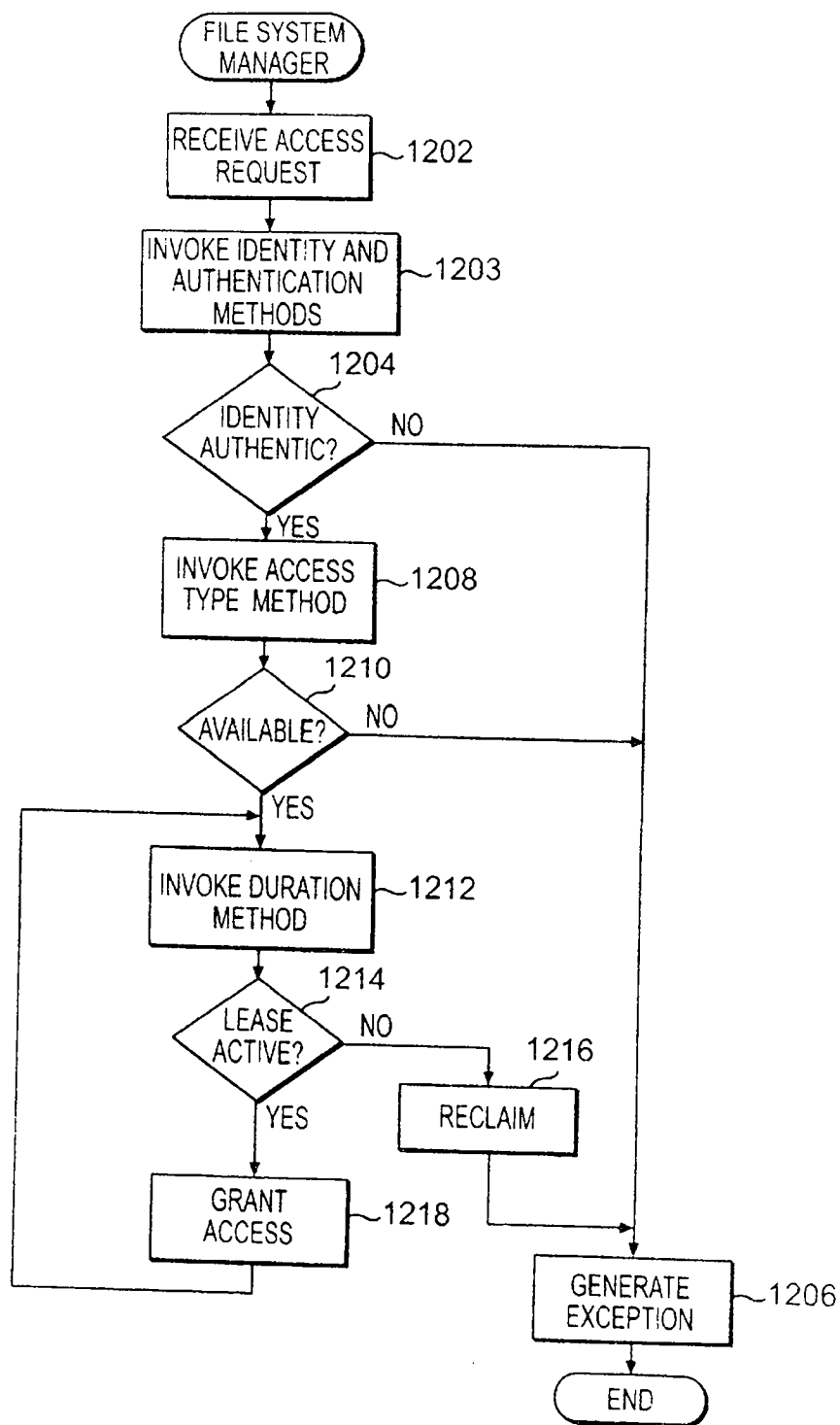


FIG. 9

**FIG. 10**

**FIG. 11A**

**FIG. 11B**

**FIG. 12**

486



European Patent Office
Postbus 5818
2280 HV RIJSWIJK
NETHERLANDS
Tel.: +31 70 340 2040
Fax: +31 70 340 3016

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets



Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date	29-08-2007
------	------------

Reference EP27314-034/df	Application No./Patent No. 03009717.4 - 1244 / 1376977
Applicant/Proprietor MICROSOFT CORPORATION	

Correction of errors in the European patent specification

I. ☒ Your request of 19.07.2007 for correction of errors is accepted. The errors notified are

☒ errors which do not affect the substance of the patent specification and therefore do not warrant publication of a printed version of the corrected specification or publication of a note concerning the correction in the Patent Bulletin. The file copy of this specification has, however, been corrected. The extent of the corrections is shown on the attached specification, which bears the Office seal.

☐ errors which distort the meaning and warrant a newly printed version of the specification/title page. A note concerning the correction will be published in Section II.12 of the Patent Bulletin. The extent and type of the corrections is shown on attached copy of the specification/title page.

II. ☐ The specification has been found to contain errors which distort the meaning. These are being corrected by the Office of its own motion. A note concerning the corrections will be published in Section II.12 of the Patent Bulletin. The extent and type of the corrections is shown on the attached copy of the specification/title page.

☒ Copy of the manually corrected patent specification enclosed

☐ After publication, approximately two months after notification of this communication, a CORRIGENDUM can be downloaded free of charge from the EPO publication server

<https://publications.european-patent-office.org>

or ordered from the Vienna sub-office upon payment of a fee.

The publication of the corrigendum will be announced in the European Patent Register.

For the Examining Division



Legendre, Marie



(11) **EP 1 376 977 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
15.08.2007 Bulletin 2007/33

(51) Int Cl.:
G06F 9/46 (2006.01) H04L 29/06 (2006.01)

(21) Application number: **03009717.4**

(22) Date of filing: **30.04.2003**

(54) **Identity-based distributed computing for device resources**

Verteiltes Rechnersystem für Vorrichtungsressourcen basierend auf Identität

Système informatique répartie pour ressources de dispositif basé sur l'identité

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IT LI LU MC NL PT RO SE SI SK TR**

(30) Priority: **28.06.2002 US 187221**

(43) Date of publication of application:
02.01.2004 Bulletin 2004/01

(73) Proprietor: **MICROSOFT CORPORATION**
Redmond, WA 98052 (US)

(72) Inventors:
• **Alles, David**
Seattle, Washington 98103 (US)

• **Moore, George M.**
Issaquah, Washington 98029 (US)

(74) Representative: **Grünecker, Kinkeldey,
Stockmalr & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)**

(56) References cited:
**EP-A- 0 952 513 WO-A-00/51031
US-A- 5 640 569 US-A- 5 898 870
US-A- 6 078 955 US-B1- 6 247 026**

Claim 2, p.12, line 6
Patentanspruch 2, p.12,
lines 55-56
Revendication 2, p.13,
line 47



EP 1 376 977 B1

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

482

the mobile computing device performing a printing task including providing printing data via the another computing device to the printer driver.

2. The method of claim 1 further comprising

communicating (600) a request to use the ~~resource~~ *printer driver* from the mobile computing device to the another computing device.

3. The method of claim 1 or 2 further comprising:

communicating identity information from the mobile computing device to the another computing device.

4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).

5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

6. A computer-readable medium having computer-executable instructions to perform the method of one of claim 1 to 5.

7. A system in a computing environment for allowing a mobile computing device not having a printer driver to print on a printer, the system comprising:

a first computing device (212, 404,) associated with at least one identity, the first device having a printer driver and configured to provide access to the printer driver by at least a second device (202, 400) associated with the identity, the first computing device providing to the second device information (209B) about the printer driver according to a resource schema associated with the at least one identity; the second device (202, 400), the second device being for mobile computing device configured to request access to the printer driver; and the second device performing a printing task including accessing the printer driver on the first device to perform at least part of the printing task.

8. The system of claim 7 further comprising a third computing device (402,) connected to the first computing device, wherein the second device accesses the printer driver to perform an operation with the third computing device, wherein the third computing device comprises the printer . (402).

Patentansprüche

1. Verfahren eines mobilen Computergeräts zum Drucken in einer Computerumgebung, wobei das Verfahren erlaubt, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das Verfahren umfasst:

Entdecken, durch das mobile Computergerät (202, 400), das mit mindestens einer Identität assoziiert ist, eines anderen Computergeräts (212, 404), das mit der mindestens einen Identität assoziiert ist, und das mit einem Druckertreiber (406) für den besagten Drucker ausgestattet ist;

Erlangen, durch das mobile Computergerät, von Daten (209B), die das andere Computergerät betreffen, wobei die Daten Druckertreiberinformationen des anderen Computergeräts einschließen, die gemäß einem Schema, das das mobile Computergerät versteht, formatiert sind;

Interpretieren, durch das mobile Computergerät, der Daten, um auf den Druckertreiber des anderen Computergeräts zuzugreifen; und

Ausführen, durch das mobile Computergerät, eines Druckauftrags, einschließlich des Bereitstellens von Druckdaten zu dem Druckertreiber über das andere Computergerät.

2. Verfahren nach Anspruch 1, das des Weiteren umfasst:

Übermitteln (600) einer Anfrage von dem mobilen Computergerät zu dem anderen Computergerät, um die ~~Resource~~ *den Druckertreiber* zu verwenden.

3. Verfahren nach Anspruch 1 oder 2, das des Weiteren umfasst:



158

483

EP 1 376 977 B1

Übermitteln von Identitätsinformationen von dem mobilen Computergerät zu dem anderen Computergerät.

4. Verfahren nach Anspruch 3, wobei die Identitätsinformationen Berechtigungsnachweisen (207A, 207B) entsprechen.
5. Verfahren nach Anspruch 4, wobei die Berechtigungsnachweise durch einen Authentifizierungsdienst (220) bereitgestellt werden.
6. Computer-lesbarer Datenträger, der Computer-ausführbare Instruktionen aufweist, um das Verfahren eines der Ansprüche 1 bis 5 auszuführen.
7. System in einer Computerumgebung zum Erlauben, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das System umfasst:

Ein erstes Computergerät (212, 404), das mit mindestens einer Identität assoziiert ist, wobei das erste Gerät einen Druckertreiber aufweist und eingerichtet ist zum Bereitstellen eines Zugriffs auf den Druckertreiber durch mindestens ein zweites Gerät (202, 400), das mit der mindestens einen Identität assoziiert ist, und wobei das erste Computergerät dem zweiten Gerät Informationen (209B) über den Druckertreiber gemäß einem Ressourcen-Schema zugehörig zu der mindestens einen Identität bereitstellt;
das zweite Gerät (202, 400), wobei das zweite Gerät das mobile Computergerät ist, das eingerichtet ist, einen Zugriff auf den Druckertreiber anzufordern; und
wobei das zweite Gerät einen Druckauftrag ausführt, der das Zugreifen auf den Druckertreiber auf dem ersten Gerät einschließt, um mindestens einen Teil des Druckauftrags auszuführen.

8. System nach Anspruch 7, das des Weiteren ein drittes Computergerät (402), das mit dem ersten Computergerät verbunden ist, umfasst, wobei das zweite Computergerät auf den Druckertreiber zugreift, um einen Arbeitsvorgang mit dem dritten Computergerät auszuführen, und wobei das dritte Computergerät den Drucker (402) umfasst.

Revendications

1. Procédé d'impression pour dispositif informatique portable dans un environnement informatique pour permettre à un dispositif informatique portable ne possédant pas de pilote d'imprimante d'imprimer sur une imprimante, le procédé comprenant :

le dispositif informatique portable (202, 400) associé à au moins une identité, découvrant un autre dispositif informatique (212, 404) associé à l'identité, équipé d'un pilote d'imprimante (406) pour ladite imprimante ;
le dispositif informatique portable obtenant des données (209B) liées à l'autre dispositif informatique, les données comprenant des informations de pilote d'imprimante de l'autre dispositif informatique formatées selon un schéma compris par le dispositif informatique portable ;
le dispositif informatique portable interprétant les données pour accéder au pilote d'imprimante de l'autre dispositif informatique ; et
le dispositif informatique portable effectuant une tâche d'impression comprenant la fourniture de données d'impression par l'intermédiaire de l'autre dispositif informatique au pilote d'imprimante.

2. Procédé selon la revendication 1, comprenant en outre :

la communication (600) d'une demande pour utiliser ~~la ressource~~ *le pilote d'imprimante* provenant du dispositif informatique portable, à destination de l'autre dispositif informatique.

3. Procédé selon la revendication 1 ou 2, comprenant en outre :

la communication d'informations d'identité, provenant du dispositif informatique portable à destination de l'autre dispositif informatique.

4. Procédé selon la revendication 3, dans lequel les informations d'identité correspondent à des justificatifs d'identité (207A, 207B).

5. Procédé selon la revendication 4, dans lequel les justificatifs d'identité sont fournis par un service d'authentification.



159



European Patent Office
Postbus 5818
2280 HV RIJSWIJK
NETHERLANDS
Tel.: +31 70 340 2040
Fax: +31 70 340 3016

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

484



Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date
10-08-2007

Reference EP27314-034/df	Application No./Patent No. 03009717.4 - 1244 / 1376977
Applicant/Proprietor MICROSOFT CORPORATION	

BRIEF COMMUNICATION

- Subject: ☐ Your letter of
☐ Our telephone conversation of
☐ Communication of
☒ Form 2050 of 03.08.2007.
- Enclosure(s): ☐ Letter from the proprietor of the patent of
☐ Letter from the opponent of
☐ Copy(copies)
☒ Communication:

By mistake we sent you Form 2050 of 03.08.2007.. Please
consider this Form as null and void.

Please take note.

For the Opposition Division



☐ Registered letter

EPO Form 2911 05.02 02.08.07

AG01198

160

485



European Patent Office
Postbus 5818
2280 HV RIJSWIJK
NETHERLANDS
Tel.: +31 70 340 2040
Fax: +31 70 340 3016

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets



Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date

03-08-2007

Reference EP27314-034/df	Application No./Patent No. 03009717.4 - 1244 / 1376977
Applicant/Proprietor MICROSOFT CORPORATION	

Your request of 19.07.2007 for correction after grant

1. Corrections under Rule 88 EPC are only possible where proceedings (for example, opposition proceedings) are pending before the EPO.

Your request for correction under Rule 88 EPC cannot be accepted, as the grant procedure has been completed and no further proceedings are pending.

However, should opposition proceedings be instituted, you will be able to file a new request for correction under Rule 88 EPC.

2. The correction cannot be allowed under Rule 89 EPC either, since a decision to grant a European patent has been issued and this decision corresponds to the intention of the Examining Division.

For the Examining Division



Groeneveld, Aad

161



European Patent Office
Postbus 5818
2280 HV RIJSWIJK
NETHERLANDS
Tel.: +31 70 340 2040
Fax: +31 70 340 3016

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

486



Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date

19.07.07

Reference EP27314-034/df	Application No./Patent No. 03009717.4 - 1244 / 1376977
Applicant/Proprietor MICROSOFT CORPORATION	

Decision to grant a European patent pursuant to article 97(2) EPC

Following examination of European patent application No. 03009717.4 a European patent with the title and the supporting documents indicated in the communication pursuant to Rule 51(4) EPC dated 09.03.07 is hereby granted in respect of the designated Contracting States.

Patent No. : 1376977
Date of filing : 30.04.03
Priority claimed : 28.06.02/USA 187221

Designated Contracting States
and Proprietor(s) : AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL
PT RO SE SI SK TR
MICROSOFT CORPORATION
One Microsoft Way
Redmond, WA 98052/US

This decision will take effect on the date on which the European Patent Bulletin mentions the grant (Art. 97(4) and (5) EPC).

The mention of the grant will be published in European Patent Bulletin 07/33 of 15.08.07.

Examining Division

Goya Alonso J

Peeters D

Kesting V



Registered letter

EPO Form 2006A 07.02 13.07.07

to EPO postal service: 13.07.07

162

ANMERKUNG ZUR ENTSCHEIDUNG ÜBER DIE ERTEILUNG EINES EUROPÄISCHEN PATENTS (EPA Form 2006)

1. **EPA Informationsbroschüre "Nationales Recht zum EPÜ"**
Diese Broschüre enthält nützliche Informationen zu den formalen Erfordernissen und den Handlungen, die vor den Patentbehörden der Vertragsstaaten vorzunehmen sind, um Rechte in diesen Staaten zu erlangen. Da diese Handlungen einem ständigen Wandel unterworfen sind, sollte immer nur die neueste Ausgabe der Broschüre benutzt werden. Nachträgliche Informationen werden im Amtsblatt veröffentlicht.
2. **Übersetzung der europäischen Patentschrift nach Artikel 65(1) des Europäischen Patentübereinkommens**
Sie werden erneut darauf hingewiesen, dass bestimmte Vertragsstaaten nach Artikel 65(1) EPÜ eine Übersetzung der europäischen Patentschrift verlangen; hierauf wird in der Mitteilung gemäss Regel 51(6) verwiesen. **Die Nichteinreichung dieser Übersetzung kann zur Folge haben, dass das Patent in dem betreffenden Staat/in den betreffenden Staaten als von Anfang an nicht eingetreten gilt.** Weitere Einzelheiten entnehmen Sie bitte der oben genannten Broschüre.
3. **Zahlung von Jahresgebühren für europäische Patente**
Nach Artikel 141 EPU können "nationale" Jahresgebühren für das europäische Patent für die Jahre erhoben werden, die an das Jahr anschliessen, in dem der Hinweis auf die Erteilung des europäischen Patents im "Europäischen Patentblatt" bekanntgemacht wird. Weitere Einzelheiten entnehmen Sie bitte der oben genannten Broschüre.

NOTE RELATING TO THE DECISION TO GRANT A EUROPEAN PATENT (EPO Form 2006)

1. **EPO Information Brochure "National law relating to the EPC"**
This brochure provides useful information regarding formal requirements and the steps to be taken before the patent authorities of the Contracting States in order to acquire rights in those states. Since the necessary steps are subject to change the latest edition of the brochure should always be used. Subsequent information is published in the Official Journal.
2. **Translation of the European patent specification under Article 65(1) of the European Patent Convention**
Your attention is again drawn to the requirements regarding translation of the European patent specification laid down by a number of Contracting States under Article 65(1) EPC, to which reference is made in the communication under Rule 51(6). **Failure to supply such translation(s) may result in the patent being deemed to be void "ab initio" in the State(s) in question.** For further details you are recommended to consult the above-mentioned brochure.
3. **Payment of renewal fees for European patents**
Under Article 141 EPC "national" renewal fees in respect of a European patent may be imposed for the years which follow that in which the mention of the grant of the European patent is published in the "European Patent Bulletin". For further details you are recommended to consult the above-mentioned brochure.

REMARQUE RELATIVE A LA DECISION DE DELIVRANCE D'UN BREVET EUROPEEN (OEB Form 2006)

1. **Brochure d'information de l'OEB "Droit national relatif à la CBE"**
Cette brochure fournit d'utiles renseignements sur les conditions de forme requises et sur les actes à accomplir auprès des offices de brevet des Etats contractants aux fins d'obtenir des droits dans les Etats contractants. Etant donné que les actes indispensables sont susceptibles de modifications, il serait bon de toujours consulter la dernière édition de la brochure. Toute information ultérieure est publiée au Journal Officiel.
2. **Traduction du fascicule du brevet européen en vertu de l'article 65(1) de la Convention sur le brevet européen**
Votre attention est de nouveau attirée sur l'obligation faite par certains Etats contractants, en vertu de l'article 65(1) CBE, de fournir une traduction du fascicule du brevet européen, à laquelle il est fait référence dans la notification établie conformément à la règle 51(6). **Si la(les) traduction(s) n'est(ne sont) pas fournie(s), le brevet européen peut, dès l'origine, être réputé sans effet dans cet(ces) Etat(s).** Pour plus de détails, nous vous renvoyons à la brochure susmentionnée.
3. **Paiement des taxes annuelles pour le brevet européen**
Conformément à l'article 141 CBE, les taxes annuelles "nationales" dues au titre du brevet européen peuvent être perçues pour les années suivant celle au cours de laquelle la mention de la délivrance du brevet européen est publiée au "Bulletin européen des brevets". Pour plus de détails, nous vous renvoyons à la brochure susmentionnée.

488

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER
ANWALTSSOZIOZETÄT

EPO - Munich
80

19. Juli 2007.

GKS & S MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

European Patent Office
Erhardtstraße 27

80298 München

RECHTSANWÄLTE
LAWYERS

MÜNCHEN
DR. HEINRICH EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRODER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANKÉ, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA BERTRAM, LL.M.
KARIN LOCHNER
BABETT ERTLE
CHRISTINE NEUMERL
SABINE PRÖCKNER
CORNELIA SCHMITT
BERNHARD MEHNERT, DIPLOM.
PETRA LÜBBE
DR. SONJA BROTJE
SONJA SCHÄFFLER
DR. HOLGER GAUSS
DR. NICOLAS SCHMITZ

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER M. JAKOB
HANS HILGERS
ANNELE EHNOLD
THOMAS SCHUSTER
DR. KLARA GOLDBACH
MARTIN AUFENANGER
GOTTFRIED KUTZSCH
DR. HEIKE VOGELSANG-WENKE
REINHARD KNAUER
DIETMAR KUHLE
DR. FRANZ-JOSEF ZIMMER
BETTINA K. REICHELT
DR. ANTON K. PFÄU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (U.G.P.A.) M.S.
BERND ROTHMEL
THOMAS W. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS EICKELKAMP
DR. CARLO TORTI
DR. PETER MÜLLENY
DR. MORITZ HOFFE
*EUROPEAN PATENT ATTORNEY

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED BÖNING
DR. PATRICK ERK, M.S. (M.D.)
KÖLN
DR. MARTIN DROPMANN
CHEMNITZ
DR. CARMEN STEINGER
ALICANTE
WOLFGANG MEISTER
DR. ANDREAS TORKA
*RECHTSANWALT
OF COUNSEL
DR. HENNING MEYER-PLATH
DR. WILFRIED STOCKMAIR
(1998)

IHR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.

DATUM / DATE

EP27314NN904awe

July 19, 2007

European Patent Application No.: 03 009 717.4
Applicant: MICROSOFT CORPORATION

In response to the Communication pursuant to Rule 51 (4) EPC dated March 09, 2007 and our response settling said Communication dated July 03, 2007:

Correction under Rule 88 (obvious error):

The following documents are submitted herewith:

- amended version of the claim 2 (the word resource was replaced by the word printer driver)
- amended French and German translations of the claims

The EPO is herewith kindly requested to change the official file accordingly.

Jens Hammer
- Patent Attorney -

Endis.

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
<http://www.grunecker.de>
e-mail: info@grunecker.de

Deutsche Bank München
Konto / Account: 175 1734
BIC / Bank code: 700 700 10
SWIFT / BIC-code: DEUT DE MM
IBAN: DE47 7007 0010 0175 1734 00

15.02.07

Eichlly

Application No.: ~~03-009-717.4-1244~~
 Applicant: MICROSOFT CORPORATION
 Our Ref.: EP27314TE109
 Date: ~~January 15, 2007~~

AUXILIARY
 MAIN REQUEST

^{computing}
 mobile device printing

of allowing a mobile device ^{<->} not having a printer driver to print on a printer,

1. A method in a computing environment for ~~facilitating accessing the computing resources of devices by other devices~~, the method comprising:
~~at a first computing device (202, 400, 500) associated with at least one identity,~~
^{the mobile device}
~~discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity,~~
^{another}
~~equipped with a printer driver (406) for said printer;~~
^{the mobile device}
~~at the first device,~~ obtaining data (209B) related to the ~~second~~ ^{another computing} device, the data including ~~resource~~ ^{printer driver} information of the ~~second~~ ^{another computing} device formatted according to a schema understood by the ~~first~~ ^{mobile <->} device;
~~the first device interpreting the data to access a resource (R_{BT}, R_{BT}) of the second device; and~~
^{mobile <->} the first device interpreting the data to access ~~a resource (R_{BT}, R_{BT})~~ ^{the printer driver} of the ~~second~~ ^{another computing} device; and
^{mobile <->} the first device performing a ~~computing~~ ^{printing} task including ~~accessing the resource of the another second device to perform at least part of the computing task.~~ ^{providing printing data via the printer driver.}
2. The method of claim 1 further comprising:
^{printer driver} communicating (600) a request to use the ~~resource~~ ^{mobile <->} from the first device to the ~~another second device.~~ ^{another computing}
3. The method of claim 1 or 2 further comprising:
^{mobile <->} communicating identity information from the first device to the ~~second~~ ^{another computing} device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

<Computing>

Gichilly

6. A computer-readable medium having computer-executable instructions to perform ~~the~~ ^{of one of claims 1 to 5:} a method ~~for facilitating accessing the computing resources of devices by other~~ devices, the method comprising:
- at a first computing device (202, 400, 500) associated with at least one identity:
- locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
- selecting a resource of a second device from the information, the second device being associated with the at least one identity;
- accessing the resource of the second device; and
- performing a computing task including accessing the resource of the second device ~~to perform at least part of the computing task.~~
7. A system in a computing environment for ^{allowing a mobile computing device not having a printer driver to print on a printer,} ~~facilitating accessing the computing resources of devices by other devices~~, the system comprising:
- ~~a first~~ computing device (212, 404, ~~504~~) associated with at least one identity, the first device ^{having a printer driver} ~~including a resource (R_{B1} - R_{Bn})~~ and configured to provide access to the ~~resource~~ ^{printer driver} by at least ~~one other~~ device (202, 400, ~~500~~) associated with the identity, ^{the first computing device providing information (209B) about the resource} ~~the first computing device providing information (209B) about the resource~~ ^{by a second printer driver} according to a resource schema associated with the at least one identity;
- ~~a~~ second device (202, 400, ~~500~~), the second device ^{being a mobile computing device} ~~configured to request access~~ ^{to the resource} ~~to the resource~~; and
- the second device performing a ^{printing} ~~computing~~ task including accessing the ^{printer driver} ~~resource~~ on the first device to perform at least part of the ^{printing} ~~computing~~ task.
- ~~8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).~~
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
- ~~11. The system of claim 7 wherein the resource comprises an input method.~~
- ① to the second device

Lidell

- ~~12. The system of claim 7 wherein the resource comprises an output method.~~
13. The system of ~~one of claims 7 to 12~~ further comprising a third computing device (402, ~~502~~) ^{print driver} connected to the first computing device, wherein the second device accesses the ~~resource~~ to perform an operation with the third computing device,
- ~~14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.~~ *the*
- ~~15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.~~
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
- ~~17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:~~
- ~~a first field for identifying a type of resource;~~
 - ~~a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;~~
 - ~~a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.~~
18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes ~~information corresponding to total size data of the storage resource.~~

Lidley

- ~~20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.~~
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
- ~~29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).~~

REVENDEICATIONS

1. Procédé d'impression pour dispositif informatique portatif dans un environnement informatique pour permettre à un dispositif informatique portatif ne possédant pas de pilote d'imprimante d'imprimer sur une imprimante, le procédé comprenant :
 - le dispositif informatique portatif (202, 400) associé à au moins une identité, découvrant un autre dispositif informatique (212, 404) associé à l'identité, équipé d'un pilote d'imprimante (406) pour ladite imprimante ;
 - le dispositif informatique portatif obtenant des données (209B) liées à l'autre dispositif informatique, les données comprenant des informations de pilote d'imprimante de l'autre dispositif informatique formatées selon un schéma compris par le dispositif informatique portatif ;
 - le dispositif informatique portatif interprétant les données pour accéder au pilote d'imprimante de l'autre dispositif informatique ; et
 - le dispositif informatique portatif effectuant une tâche d'impression comprenant la fourniture de données d'impression par l'intermédiaire de l'autre dispositif informatique au pilote d'imprimante.
2. Procédé selon la revendication 1, comprenant en outre :
 - la communication (600) d'une demande pour utiliser le pilote d'imprimante, provenant du dispositif informatique portatif, à destination de l'autre dispositif informatique.
3. Procédé selon la revendication 1 ou 2, comprenant en outre :
 - la communication d'informations d'identité,

provenant du dispositif informatique portatif à destination de l'autre dispositif informatique.

4. Procédé selon la revendication 3, dans lequel les informations d'identité correspondent à des justificatifs d'identité (207A, 207B).

5. Procédé selon la revendication 4, dans lequel les justificatifs d'identité sont fournis par un service d'authentification (220).

6. Support lisible par ordinateur possédant des instructions pouvant être exécutées par ordinateur pour exécuter le procédé selon l'une des revendications 1 à 5.

7. Système dans un environnement informatique pour permettre à un dispositif informatique portatif ne possédant pas de pilote d'imprimante d'imprimer sur une imprimante, le système comprenant :

un premier dispositif informatique (212, 404) associé à au moins une identité, le premier dispositif possédant un pilote d'imprimante et étant configuré pour fournir un accès au pilote d'imprimante par au moins un deuxième dispositif (202, 400) associé à l'identité, le premier dispositif informatique fournissant au deuxième dispositif des informations (209B) concernant le pilote d'imprimante selon un schéma de ressource associé à l'identité ;

le deuxième dispositif (202, 400), celui-ci étant le dispositif informatique portatif configuré pour demander l'accès au pilote d'imprimante ; et

le deuxième dispositif effectuant une tâche d'impression comprenant l'accès au pilote d'imprimante sur le premier dispositif pour effectuer au moins une partie de la tâche d'impression.

8. Système selon la revendication 7, comprenant en outre un troisième dispositif informatique (402) relié au premier dispositif informatique, dans lequel

EP 03 009 717.4

3

le deuxième dispositif accède au pilote d'imprimante pour effectuer une opération avec le troisième dispositif informatique, dans lequel le troisième dispositif informatique comprend l'imprimante (402).

EP 03 009 717.4

Patentansprüche

1. Verfahren eines mobilen Computergeräts zum Drucken in einer Computerumgebung, wobei das Verfahren erlaubt, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das Verfahren umfasst:
Endecken, durch das mobile Computergerät (202, 400), das mit mindestens einer Identität assoziiert ist, eines anderen Computergeräts (212, 404), das mit der mindestens einen Identität assoziiert ist, und das mit einem Druckertreiber (406) für den besagten Drucker ausgestattet ist;
Erlangen, durch das mobile Computergerät, von Daten (209B), die das andere Computergerät betreffen, wobei die Daten Druckertreiberinformationen des anderen Computergeräts einschließen, die gemäß einem Schema, das das mobile Computergerät versteht, formatiert sind;
Interpretieren, durch das mobile Computergerät, der Daten, um auf den Druckertreiber des anderen Computergeräts zuzugreifen; und
Ausführen, durch das mobile Computergerät, eines Druckauftrags, einschließlich des Bereitstellens von Druckdaten zu dem Druckertreiber über das andere Computergerät.
2. Verfahren nach Anspruch 1, das des Weiteren umfasst:
Übermitteln (600) einer Anfrage, den Druckertreiber zu verwenden, von dem mobilen Computergerät zu dem anderen Computergerät.
3. Verfahren nach Anspruch 1 oder 2, das des Weiteren umfasst:
Übermitteln von Identitätsinformationen von dem mobilen Computergerät zu dem anderen Computergerät.
4. Verfahren nach Anspruch 3, wobei die Identitätsinformationen Berechtigungsnachweisen (207A, 207B) entsprechen.
5. Verfahren nach Anspruch 4, wobei die Berechtigungsnachweise durch einen Authentifizierungsdienst (220) bereitgestellt werden.

6. Computer-lesbarer Datenträger, der Computer-ausführbare Instruktionen aufweist, um das Verfahren eines der Ansprüche 1 bis 5 auszuführen.
7. System in einer Computerumgebung zum Erlauben, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das System umfasst:

Ein erstes Computergerät (212, 404), das mit mindestens einer Identität assoziiert ist, wobei das erste Gerät einen Druckertreiber aufweist und eingerichtet ist zum Bereitstellen eines Zugriffs auf den Druckertreiber durch mindestens ein zweites Gerät (202, 400), das mit der mindestens einen Identität assoziiert ist, und wobei das erste Computergerät dem zweiten Gerät Informationen (209B) über den Druckertreiber gemäß einem Ressourcen-Schema zugehörig zu der mindestens einen Identität bereitstellt;

das zweite Gerät (202, 400), wobei das zweite Gerät das mobile Computergerät ist, das eingerichtet ist, einen Zugriff auf den Druckertreiber anzufordern; und

wobei das zweite Gerät einen Druckauftrag ausführt, der das Zugreifen auf den Druckertreiber auf dem ersten Gerät einschließt, um mindestens einen Teil des Druckauftrags auszuführen.
8. System nach Anspruch 7, das des Weiteren ein drittes Computergerät (402), das mit dem ersten Computergerät verbunden ist, umfasst, wobei das zweite Computergerät auf den Druckertreiber zugreift, um einen Arbeitsvorgang mit dem dritten Computergerät auszuführen, und wobei das dritte Computergerät den Drucker (402) umfasst.

498

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER
ANWALTSSOZIELTÄT

GKS & S MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

European Patent Office
Erhardtstraße 27

80298 München

RECHTSANWÄLTE
LAWYERS

MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRÖDER, LL.M.
CHRISTIAN NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANKS, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA BERTRAM, LL.M.
KARIN LOCHNER
BABETT ERTLE
CHRISTINE NEUHIERL
SABINE PRÜCKNER
CORNELIA SCHMITT
BERNHARD MEINERT, DIPL.ING.
PETRA LÜBBE
DR. SONJA BRÖTJE
SONJA SCHÄFFLER
DR. HOLGER GAUSS
DR. NICOLAS SCHMITZ

EPO - Munich
86

03. April 2007
PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOB
HANS MILGERS
ANNELE EHNOLD
THOMAS SCHUSTER
DR. KLAUS GOLDBACH
MARTIN KUFENVANGER
GOTTFRIED KUTZSCH
DR. HEIKE VOGELSONG-WENKE
REINHARD KRAUER
DIETMAR KUHLE
DR. FRANZ-JOSEF ZIMMER
BETTINA K. REICHELT
DR. ANTON K. PFALL
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M. S. (U.S.P.A.) M. S.
BERND ROTHMEIER
THOMAS W. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS FICKELNAMP
DR. CARLO TORTI
DR. PETER WILTENNY
DR. MORITZ HOFFE
EUROPEAN PATENT ATTORNEYS

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED RÖNING
DR. PATRICK ERK M. S. (M.F.)
KÖLN
DR. MARTIN DROPMANN
CHEMNITZ
DR. CARMEN STEINIGER
ALICANTE
WOLFGANG MEISTER
DR. ANDREAS TORKA
RECHTSANWALT
OF COUNSEL
DR. HENNING MEYER-FATH
DR. WILFRIED STOCKMAIR
1996

IHR ZEICHEN / YOUR REF

UNSER ZEICHEN / OUR REF

EP27314NN904awe

DATUM / DATE

July 3, 2007

European Patent Application No. 03 009 717.4
Applicant: Microsoft Corporation

In response to the Communication under Rule 51(4) EPC dated March 09, 2007:

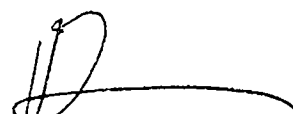
The following documents are submitted herewith:

- French and German translations of the claims
- EPO Form 1010

Please debit our account no. 28 00 04 37 with the official printing fee of EUR 948,--.

The above amount also covers the additional printing fee of EUR 198,-- for 18 additional pages.

We herewith request one copy of the printed patent specification when issuing the certificate.


Dr. F.-J. Zimmer
- Patent Attorney -

Zur Kasse
(A) € 948,-

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
<http://www.grunecker.de>
e-mail: info@grunecker.de

Deutsche Bank München
Konto / Account: 175 1734
BLZ / Bank code: 700 700 10
SWIFT / BIC-code: DEUT DE MM
IBAN: DE47 7007 0010 0175 1734 00

174

REVENDEICATIONS

1. Procédé d'impression pour dispositif informatique portatif dans un environnement informatique pour permettre à un dispositif informatique portatif ne possédant pas de pilote d'imprimante d'imprimer sur une imprimante, le procédé comprenant :

le dispositif informatique portatif (202, 400) associé à au moins une identité, découvrant un autre dispositif informatique (212, 404) associé à l'identité, équipé d'un pilote d'imprimante (406) pour ladite imprimante ;

le dispositif informatique portatif obtenant des données (209B) liées à l'autre dispositif informatique, les données comprenant des informations de pilote d'imprimante de l'autre dispositif informatique formatées selon un schéma compris par le dispositif informatique portatif ;

le dispositif informatique portatif interprétant les données pour accéder au pilote d'imprimante de l'autre dispositif informatique ; et

le dispositif informatique portatif effectuant une tâche d'impression comprenant la fourniture de données d'impression par l'intermédiaire de l'autre dispositif informatique au pilote d'imprimante.

2. Procédé selon la revendication 1, comprenant en outre :

la communication (600) d'une demande pour utiliser la ressource, provenant du dispositif informatique portatif, à destination de l'autre dispositif informatique.

3. Procédé selon la revendication 1 ou 2, comprenant en outre :

la communication d'informations d'identité,

provenant du dispositif informatique portatif à destination de l'autre dispositif informatique.

4. Procédé selon la revendication 3, dans lequel les informations d'identité correspondent à des justificatifs d'identité (207A, 207B).

5. Procédé selon la revendication 4, dans lequel les justificatifs d'identité sont fournis par un service d'authentification (220).

6. Support lisible par ordinateur possédant des instructions pouvant être exécutées par ordinateur pour exécuter le procédé selon l'une des revendications 1 à 5.

7. Système dans un environnement informatique pour permettre à un dispositif informatique portatif ne possédant pas de pilote d'imprimante d'imprimer sur une imprimante, le système comprenant :

un premier dispositif informatique (212, 404) associé à au moins une identité, le premier dispositif possédant un pilote d'imprimante et étant configuré pour fournir un accès au pilote d'imprimante par au moins un deuxième dispositif (202, 400) associé à l'identité, le premier dispositif informatique fournissant au deuxième dispositif des informations (209B) concernant le pilote d'imprimante selon un schéma de ressource associé à l'identité ;

le deuxième dispositif (202, 400), celui-ci étant le dispositif informatique portatif configuré pour demander l'accès au pilote d'imprimante ; et

le deuxième dispositif effectuant une tâche d'impression comprenant l'accès au pilote d'imprimante sur le premier dispositif pour effectuer au moins une partie de la tâche d'impression.

8. Système selon la revendication 7, comprenant en outre un troisième dispositif informatique (402) relié au premier dispositif informatique, dans lequel

EP 03 009 717.4

3

le deuxième dispositif accède au pilote d'imprimante pour effectuer une opération avec le troisième dispositif informatique, dans lequel le troisième dispositif informatique comprend l'imprimante (402).

EP 03 009 717.4

Patentansprüche

1. Verfahren eines mobilen Computergeräts zum Drucken in einer Computerumgebung, wobei das Verfahren erlaubt, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das Verfahren umfasst:

Endecken, durch das mobile Computergerät (202, 400), das mit mindestens einer Identität assoziiert ist, eines anderen Computergeräts (212, 404), das mit der mindestens einen Identität assoziiert ist, und das mit einem Druckertreiber (406) für den besagten Drucker ausgestattet ist;

Erlangen, durch das mobile Computergerät, von Daten (209B), die das andere Computergerät betreffen, wobei die Daten Druckertreiberinformationen des anderen Computergeräts einschließen, die gemäß einem Schema, das das mobile Computergerät versteht, formatiert sind;

Interpretieren, durch das mobile Computergerät, der Daten, um auf den Druckertreiber des anderen Computergeräts zuzugreifen; und

Ausführen, durch das mobile Computergerät, eines Druckauftrags, einschließlich des Bereitstellens von Druckdaten zu dem Druckertreiber über das andere Computergerät.
2. Verfahren nach Anspruch 1, das des Weiteren umfasst:

Übermitteln (600) einer Anfrage von dem mobilen Computergerät zu dem anderen Computergerät, um die Ressource zu verwenden.
3. Verfahren nach Anspruch 1 oder 2, das des Weiteren umfasst:

Übermitteln von Identitätsinformationen von dem mobilen Computergerät zu dem anderen Computergerät.
4. Verfahren nach Anspruch 3, wobei die Identitätsinformationen Berechtigungsnachweisen (207A, 207B) entsprechen.
5. Verfahren nach Anspruch 4, wobei die Berechtigungsnachweise durch einen Authentifizierungsdienst (220) bereitgestellt werden.

6. Computer-lesbarer Datenträger, der Computer-ausführbare Instruktionen aufweist, um das Verfahren eines der Ansprüche 1 bis 5 auszuführen.
7. System in einer Computerumgebung zum Erlauben, dass ein mobiles Computergerät, das keinen Druckertreiber hat, auf einem Drucker druckt, und wobei das System umfasst:

Ein erstes Computergerät (212, 404), das mit mindestens einer Identität assoziiert ist, wobei das erste Gerät einen Druckertreiber aufweist und eingerichtet ist zum Bereitstellen eines Zugriffs auf den Druckertreiber durch mindestens ein zweites Gerät (202, 400), das mit der mindestens einen Identität assoziiert ist, und wobei das erste Computergerät dem zweiten Gerät Informationen (209B) über den Druckertreiber gemäß einem Ressourcen-Schema zugehörig zu der mindestens einen Identität bereitstellt;

das zweite Gerät (202, 400), wobei das zweite Gerät das mobile Computergerät ist, das eingerichtet ist, einen Zugriff auf den Druckertreiber anzufordern; und

wobei das zweite Gerät einen Druckauftrag ausführt, der das Zugreifen auf den Druckertreiber auf dem ersten Gerät einschließt, um mindestens einen Teil des Druckauftrags auszuführen.

8. System nach Anspruch 7, das des Weiteren ein drittes Computergerät (402), das mit dem ersten Computergerät verbunden ist, umfasst, wobei das zweite Computergerät auf den Druckertreiber zugreift, um einen Arbeitsvorgang mit dem dritten Computergerät auszuführen, und wobei das dritte Computergerät den Drucker (402) umfasst.



EPA/EPC/OEB
D-80298 München

☎ +49 89 2399-0
TX 523 656 epmu d
FAX +49 89 2399-4455

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 2

Directorate General 2

Direction Générale 2

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



Application No.
03 009 717.4 - 1244

Ref.
EP27314-034/df

Date
09.03.2007

Applicant
MICROSOFT CORPORATION

Communication under Rule 51(4) EPC

You are informed that the Examining Division intends to grant a European patent on the basis of the above application with the text and drawings as indicated below:

Auxiliary Request 1

In the text for the Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR

Description, Pages

1-3, 5-19, 21-36, 38, 39	as originally filed
4, 4b, 20, 37, 40	received on 10.01.2005 with letter of 05.01.2005
4a	received on 22.03.2006 with letter of 22.03.2006

Claims, Numbers

1-8	filed during Oral proceedings on 15.02.2007
-----	---

Drawings, Sheets

1/7-7/7	as originally filed
---------	---------------------

With the following amendments to the above-mentioned documents by the examining division:

Description, Pages	41)
Claims, Numbers	7-82)

Comments



- 1) Rule 27(1)(b) (acknowledgement of prior art document).
- 2) Article 84 EPC. Agreed with the representative during the oral proceedings.

See also comments on enclosed Form 2906.
A copy of relevant documents is enclosed

The title of the invention in the three official languages of the European Patent Office, the international patent classification, the designated Contracting States, the registered name of the applicant and the bibliographic data are shown on the attached EPO Form 2056.

You are requested within a **non-extendable** period of four months of notification of this communication

- | | | | |
|-----|--|---------------|--------|
| 1. | to file 1 set of translations of the claim(s) in the two other EPO official languages; | | |
| | | | EUR |
| 2a. | to pay the fee for grant including the fee for printing up to and including 35 pages; | | |
| | Reference 007 | | 750.00 |
| 2b. | to pay the printing fee for the 36th and each additional page; | | |
| | number of pages: 18 | | |
| | | Reference 008 | 198.00 |
| 3. | to pay the additional claim fee(s) (Rule 51(7) EPC); | | |
| | number of claims fees payable: 0 | | |
| | | Reference 016 | 0.00 |
| | | Total amount | 948.00 |

Concerning the possibility of a request for accelerated grant pursuant to Article 97(6) EPC, reference is made to OJ EPO 2001, 459.

If you do not approve the text intended for grant but wish to request amendments or corrections, the procedure described in Rule 51(5) EPC is to be followed.

If this communication is based upon an auxiliary request, and you reply within the time limit set that you maintain the main or a higher ranking request which is not allowable, the application will be refused (Article 97(1) EPC, see also Legal Advice 15/05 (rev. 02), OJ 6/2005, 357).

If the enclosed claims contain amendments proposed by the Examining Division, and you reply within the time limit set that you cannot accept these amendments, refusal of the application under Article 97(1) EPC would result in the case that agreement cannot be reached on the text for grant.

In all cases except those of the previous two paragraphs, if the grant, printing or claims fees are not paid, or the translations not filed, in due time, the European patent application will be deemed to be withdrawn (Rule 51(8) EPC).

For all payments you are requested to use EPO Form 1010 or to refer to the relevant reference number.

After publication, the European patent specification can be downloaded free of charge from the EPO publication server <https://publications.european-patent-office.org> or ordered only from the Vienna sub-office upon payment of a fee (OJ EPO 2005, 126).



Upon request in writing each proprietor will receive the certificate for the European patent **together with one copy** of the patent specification only if the request is filed within the time limit of Rule 51(4) EPC. If such request has been previously filed, it has to be confirmed within the time limit of Rule 51(4) EPC. The requested copy is free of charge. If the request is filed after expiry of the Rule 51(4) EPC time limit, the certificate will be delivered without a copy of the patent specification.

Translation of the priority document(s)

If the translation of the priority document(s), as required by Article 88(1) EPC, or the declaration according to Rule 38(5) EPC has not yet been filed, Form 2530 will be despatched separately. The translation is to be filed within the above mentioned time limit (Rule 38(5) EPC).

Note on payment of renewal fees

If a renewal fee falls due between notification of the present communication and the proposed date of publication of the mention of the grant of the European patent, publication will be effected only after the renewal fee and any additional fee have been paid (Rule 51(9) EPC).

Under Article 86(4) EPC, renewal fees are payable to the European Patent Office until the year in which the mention of the grant of the European patent is published.

Filing of translations in the Contracting States

Pursuant to Article 65(1) EPC the following Contracting States require a translation of the specification of the European patent in their/one of their official language(s) (Rule 51(10) EPC), **insofar** this specification will not be published in their/one of their official language(s)

- within **three** months of publication of the mention of such decision:

AT	AUSTRIA	GB	UNITED KINGDOM
BE	BELGIUM	GR	GREECE
BG	BULGARIA	HU	HUNGARY
CH	SWITZERLAND / LIECHTENSTEIN	IT	ITALY
CY	CYPRUS	NL	NETHERLANDS
CZ	CZECH REPUBLIC	PT	PORTUGAL
DE	GERMANY	RO	ROMANIA
DK	DENMARK	SE	SWEDEN
EE	ESTONIA	SI	SLOVENIA
ES	SPAIN	SK	SLOVAKIA
FI	FINLAND	TR	TURKEY
FR	FRANCE		

- within **six** months of publication of the mention of such decision:

IE IRELAND

The date on which the European Patent Bulletin publishes the mention of the grant of the European patent will be indicated in the decision on the grant of the European patent (EPO Form 2006).

The translation must be filed with the national Patent Offices of the Contracting or Extension States in accordance with the provisions applying thereto in the State concerned. Further details (e.g. appointment of a national representative or indication of an address for service within the country) are given in the EPO information brochure "National law relating to the EPC", and in the supplementary information published in the Official Journal of the EPO, or available on the EPO website.



Date 09.03.2007

Sheet 4

Application No.: 03 009 717.4

Failure to supply such translation to the Contracting and Extension States in time and in accordance with the requirements may result in the patent being deemed to be void ab initio in the State concerned.

Note to users of the automatic debiting procedure

Unless the EPO receives prior instructions to the contrary, the fee(s) will be debited on the last day of the period of payment. For further details see the Arrangements for the automatic debiting procedure (see Supplement to OJ EPO 2, 2002).

Examining Division:

Chairman:	Kesting, Volker
2nd Examiner:	Peeters, Dirk
1st Examiner:	Goya, Jesus



Rothenbücher, Anita
For the Examining Division
 Tel. No.: +31 70 340 - 9872

Branch at The Hague

Enclosure(s): Form 2906
 Form 2056
 53 Copies of the relevant documents
 Refusal of main request

Annex to EPO Form 2004, Communication under Rule 51(4) EPC

Bibliographical data of European patent application No. 03 009 717.4

For the intended grant of a European patent, the bibliographical data are set out below, for information:

Title of invention:

- Verteiltes Rechnersystem für Vorrichtungsressourcen basierend auf Identität
- Identity-based distributed computing for device resources
- Système informatique répartie pour ressources de dispositif basé sur l'identité

Classification: INV. G06F9/46 H04L29/06

Date of filing: 30.04.2003

Priority claimed: US / 28.06.2002 / USA187221

Contracting States*
for which fees have been paid:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL
PT RO SE SI SK TR

Extension States*
for which fees have been paid:

Applicant(s):** MICROSOFT CORPORATION
One Microsoft Way
Redmond, WA 98052
US

Inventor(s):

Alles, David
4123 Evanston Avenue North
Seattle, Washington 98103
US

Moore, George M.
5565 248th Place S.E.
Issaquah, Washington 98029
US

*) In case the time limits pursuant to Article 79(2) and Rule 85a EPC have not yet expired, **all Contracting States/Extension States** have been mentioned.

**) In case two or more applicants have designated different Contracting States, this is indicated here.



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 09.03.2007
Date

Blatt
Sheet 1
Feuille

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

1 The following documents are referred to in this annex

D1: US-A-5 898 870 (IWAKURA YOSHIYUKI ET AL) 27 April 1999 (1999-04-27)

D2: US-A-6 078 955 (KONNO CHISATO ET AL) 20 June 2000 (2000-06-20)

2 The main request of the application does not meet the requirements of Article 52(1) EPC because the subject-matter of the independent **claims 1, 6, 7 and 17** does not involve an inventive step in the sense of Article 56 EPC.

2.1 Document D1, which is considered the closest prior art, discloses in terms of **claim 1**, a method in a computing environment for facilitating accessing the computing resources of devices by other devices (abstract), the method comprising:

- at a first computing device discovering a second computing device (column 3, lines 25-50; claim 6);
- at the first device, obtaining data related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device (column 3, lines 25-46; claim 6; figure 2); and
- the first device interpreting the data to access a resource of the second device (column 3, lines 35-50; claim 6)
- the first device performing a computing task including using the resource of the second device to perform at least part of the task (column 3, lines 35-50; claim 6).

The only difference between the subject-matter of document D1 and the disclosure of D1 is:

- the first computing device is associated with at least one identity and the second computing device is associated with the at least one identity.

Regarding this only difference, the following is noted:

- This difference solves the technical problem of restricting the communication to



devices that have a certain relationship. This relationship could exist due to multiple reasons, such as the first and second computer belonging to the same network, belonging to the same organisation, etc.

- The mentioned technical problem and its solution are obvious implementation-related design options that would be known to the skilled person in the field of network design. For example in a corporate network, all computers are associated to one another because they all belong to the same corporation or the same corporate network, which inherently carry an identity.

Consequently, the subject-matter of claim 1 does not involve an inventive step in the sense of Article 56 EPC, and is therefore not allowable under Article 52(1) EPC.

2.2 **Claims 6 and 7** are a representation of claim 1 in terms of computer-readable medium having computer-executable instructions to perform the method of claim 1 and the system that implement the method of claim 1. Document D1 hints at the method of claim 1, as described in previous point 2.1, and also hints at its implementation in the form of a system (D1, claim 10), and also at the computer-executable instructions to perform such method (D5, claim 1. The method is executed in a computer). Consequently, claims 6 and 7 are also not allowable due to lack of inventive step in the sense of Article 56 EPC.

2.3 Document D2 discloses, in terms of **claim 17**, a computer-readable medium having stored thereon a resource information exchange data structure (column 4, lines 52-57; column 5, lines 62-63; figure 5), the resource information exchange data structure comprising:

- a first field for identifying a type of resource (column 6, lines 9-14; figure 5. First column in table 502);
- a second field for identifying the second device that includes a resource corresponding to the type of resource identified in the first field (column 6, lines 1-5 and figure 5. First column in table 500);
- a set of at least one third field, each third field in the set corresponding to the second field for identifying information of the second device identified by the first field (column 6, lines 17-26; figure 5: Table 505).



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 09.03.2007
Date

Blatt
Sheet 3
Feuille

Anmelde-Nr.
Application No. 03 009 117.4
Demande n°

The only difference between the subject-matter of claim 17 and the disclosure of document D2 is:

- the table being suitable for providing resource information of a second device to a first computing device.

The technical problem associated with the only difference is how to provide with the required information to a device so that it can take decisions about the place of execution of processes in a distributed system architecture. This technical problem is well known in the field of distributed architectures and has already been solved in the same way, see for example document D1 (column 3, lines 15-50. The "resource utilization-state management table" provides information to the first device about the second device and using such information the first device decides where to execute the processes).

Consequently, claim 17 does not involve an inventive step in the sense of Article 56 EPC, and is therefore not allowable under Article 52(1) EPC.

30 April 2003

512

~~IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES~~FIELD OF THE INVENTION

The invention relates generally to computer systems, and
5 more particularly to computer resources.

BACKGROUND OF THE INVENTION

In early personal computing systems, application program
had to be programmed with knowledge of various computing
10 devices in order for the application to use such devices. For
example, in order to print to a particular printer, a word
processing program had to know how to print to the printer,
and generally supplied a driver for each supported printer.
Alternatively, a printer would supply drivers with it for the
15 most popular programs. The user had to configure each program
to inform that program as to which printer was connected to
the personal computer.

Contemporary operating systems such as Microsoft Windows®
provided developers with a way to interface applications with
20 such devices regardless of the specific device driver needs
and requirements. For example, with the Microsoft Windows®
operating system a user or the device tells the operating
system which printer (or printers) is connected, after which
any application is able to communicate with the printer driver
25 via the operating system. As a result, to accomplish

188

printing, application programs generally only need to deal with a set of standardized printing functions provided by the operating system. The enormous benefits of such an abstraction cannot be understated, e.g., this abstraction enables any one of thousands of software programs to interact with any one of thousands of printers and/or other devices, without the program having to know (or otherwise supply) the details about the printer in advance.

Microsoft Corporation's .NET technology provides a somewhat similar type of abstraction regarding a user's or company's data. In general, with .NET technology, any .NET-capable program on any device can interact with a user's data via centralized schema-based services. Because of the schematization, the programs do not need specific knowledge about the details of that data, e.g., which used to be stored in proprietary file formats, but rather only needs to be able to deal with (read and write) data formatted according to one or more of the schemas.

With .NET technology, a user's (or other entity's) identity controls access to the centrally-maintained data, e.g., a user may use a .NET Inbox service keep his or her electronic mail data centrally stored (e.g., on the Internet or an intranet), and then access that data via the .NET inbox service, at any time, from any device capable of connecting to

the Internet or an intranet and having a program capable of interacting with the .NET Inbox service. The user's identity is used to securely match the user with the appropriate data. Users can also controllably allow access to data by other
5 users, e.g., a user can selectively make certain business contact information available to business associates, make personal contact information available to friends, and so on, based on the others' identities.

While .NET thus provides tremendous data-based benefits
10 to users, companies and other entities, the centralized service and centralized data are not particularly concerned with how a user uses various devices and those devices' resources. For example, devices such as a personal computer and pocket personal computer have, for quite some time, been
15 able to directly synchronize with one another. To an extent, .NET technology is another way to keep such data synchronized, although indirectly, without needing specialized and proprietary synchronization programs.

However, even with .NET, a user cannot easily use one
20 device to access the resources of another device, such as to use the computing power, storage, bandwidth, input and output mechanisms, programs and/or other resources of one device on behalf of another device, e.g., to perform a complex or better-suited task on the more powerful device. What is

needed is a common, consistent way to remotely discover and expose computing resources of various computing devices to one another for cross-resource usage.

<①>

5

SUMMARY OF THE INVENTION

~~Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.~~

10

15 Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus

20

~~abstracted into its resources, which are then programmatically~~

① EP0952513 discloses the automatic configuration of a printer server in order for a client to be able to use the printer driver located in the printer server to perform a print in a remote printer.

US-A-5 898 870 describes load sharing in a parallel computer system having a computer group constituted by a plurality of computers, and an operation management mechanism which is a computer for managing the operation of the computer group. Firstly, resource utilization target values by work are set in advance in the operation management mechanism, and subsequently the respective computers of the computer group inform the operation management mechanism of collected resource utilization states by work. In response to a request by a computer to newly start a work process, the resource utilization target value of the work process is compared with the resource utilization state thereof on a resource utilization state management table which is prepared in the initialization of the system at the operation management mechanism. Further, one of the computers is selected from the plurality of computers constituting the computer group on the basis of the result of the comparison as the optimum system and is prompted to execute the work process.

US-A-6 078 955 describes a method of controlling a computer system including a plurality of computers interconnected by a network and used by a plurality of users. The computer system includes at least a server, an administration server and a plurality of terminals. The administration server is responsible for managing all the computer resources and all the jobs performed on the computers as well as the entire user information in the computer system. The administration server collects and manages information in a system resource management table. Further, when a user logs in to the system at a terminal, the log-in information and the terminal information used for the log-in are sent to the administration server. After a processing request is sent from a terminal to the administration server, a server or terminal judged to be capable of handling a given process most efficiently is determined from the prevailing processing load of the servers in the network and the property and the location of the file and the program used for the intended process, and the place of execution of the process is notified to the particular terminal. The terminal then requests a computer corresponding with the notification to execute the process.

SUMMARY OF THE INVENTION

It is the object of the invention to provide a method and system to discover and expose computing resources of various computing devices to one another for cross-resource usage.

This object is solved by the invention as claimed in the independent claims.

Preferred embodiments are defined by the dependent claims.

[→ 4 b]

~~needed is a common, consistent way to remotely discover and
expose computing resources of various computing devices to one
another for cross-resource usage.~~

5

~~SUMMARY OF THE INVENTION~~

Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.

Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus abstracted into its resources, which are then programmatically

addressable from other devices in a composable fashion (e.g., the remote resources may be defined in such a way that they are, in turn, comprised of smaller resources contained within the larger definition of the resource) . In other words, for
5 devices to participate in an intelligent distributed computing model, various aspects of each participating device are abstracted and made programmatically addressable from other devices. A unified identity-based mechanism discovers the list of remote resources on the device. A set of one or more
10 schemas consistently describes the remote resources on the device, and a unified security model allows the owner of the device to control who has access to which resources, and to what extent.

With the present invention, desirable computing solutions
15 for a great number of example scenarios are enabled, including a set of real-time scenarios in which one user device interacts with one or more remote computing devices in real-time, and a set of scheduled or predicted scenarios that operate transparently and automatically on a user's behalf.

20 Other advantages will become apparent from the following detailed description when taken in conjunction with the drawings, in which:

519

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1A is a block diagram generally representing a computer system into which the present invention may be incorporated;

5 FIG. 1B is a block diagram generally representing another computer system into which the present invention may be incorporated;

FIG. 2 is a block diagram representing various components in a distributed computing environment in accordance with an aspect of the present invention;

10 FIG. 3 is a block diagram generally representing an architecture for discovering and exchanging resource information in accordance with an aspect of the present invention;

15 FIGS. 4 and 5 are block diagrams representing examples of how the present invention can improve a user computing experience by distributing computing operations to resources, in accordance with an aspect of the present invention;

FIG. 6 is a flow diagram representing exemplary logic and related operations performed to facilitate the sharing of resources in a distributed computing environment in accordance with an aspect of the present invention; and

195

FIG. 7 is a block diagram generally representing an example implementation including resources distributed in accordance with an aspect of the present invention.

5

DETAILED DESCRIPTION

EXEMPLARY OPERATING ENVIRONMENTS

FIGURE 1A illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one
10 example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components
15 illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for
20 use with the invention include, but are not limited to: personal computers, server computers, hand-held or laptop devices, tablet devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe

computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth, that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in local and/or remote computer storage media including memory storage devices.

With reference to FIG. 1A, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and

not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus. Note that any and all of the elements shown in computer 110 may be connected to each other by standard high-speed computer networks, including those distributed across a wide geographic area. For example, the system memory 130 may be located in a physically different location than processing unit 120, but yet defined in a composable fashion as logical computer 110.

The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 110 and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,

digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can

5 accessed by the computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data
10 signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media
15 such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read
20 only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or

program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, FIG. 1A illustrates operating system 134, application programs 135, other program modules 5 136 and program data 137.

The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1A illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile 10 magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, 15 volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is 20 typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

The drives and their associated computer storage media, discussed above and illustrated in FIG. 1A, provide storage of computer-readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1A, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as mouse, trackball or touch pad. Other input devices not shown in FIG. 1A may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected

to the system bus 121 via an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 195 and printer 196, which may be connected through an output peripheral interface 194 or the like.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1A. The logical connections depicted in FIG. 1A include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. For example, in the present invention, the computer system 110 may comprise source machine from which data is being migrated,

and the remote computer 180 may comprise the destination machine. Note however that source and destination machines need not be connected by a network or any other means, but instead, data may be migrated via any media capable of being written by the source platform and read by the destination platform or platforms.

When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160 or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1A illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

FIGURE 1B illustrates an example of another suitable operating environment (computer device 188) in which at least

part of the present invention may be implemented. As with FIG. 1A, the operating environment of FIG. 1B is only one other example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. The other computer device 188 in FIG. 1B typically includes at least some form of computer readable media. Computer-readable media can be any available media (as generally defined above) that can be accessed by the computing device 188.

FIG. 1B shows functional components of a computing device 188 such as a handheld (pocket-sized or tablet) personal digital assistant, including a processor 189, a memory 190, a display 192, and a keyboard 193 (which may be a physical or virtual keyboard). The memory 190 generally includes both volatile memory (e.g., RAM) and non-volatile memory (e.g., ROM, PCMCIA cards, and so forth). An operating system 194 is resident in the memory 190 and executes on the processor 189, such as the Windows® CE operating system from Microsoft® Corporation, or another operating system.

One or more application programs 195 are loaded into memory 190 and run on the operating system 194. Examples of applications include email programs, scheduling programs, PIM (personal information management) programs, word processing programs, spreadsheet programs, Internet browser programs, and

so forth. The handheld personal computer 188 may also include a notification manager 196 loaded in the memory 190, which executes on the processor 189. The notification manager 196 handles notification requests, e.g., from the application programs 195.

The handheld personal computer 188 has a power supply 197, which is implemented as one or more batteries. The power supply 197 may further include an external power source that overrides or recharges the built-in batteries, such as an AC adapter or a powered docking cradle.

The exemplary handheld personal computer 188 represented in FIG. 1B is shown with three types of external notification mechanisms: one or more light emitting diodes (LEDs) 198 and an audio generator 199. These devices may be directly coupled to the power supply 197 so that when activated, they remain on for a duration dictated by a notification mechanism even though the handheld personal computer processor 189 and other components might shut down to conserve battery power. The LED 198 preferably remains on indefinitely until the user takes action. Note that contemporary versions of the audio generator 199 use too much power for today's handheld personal computer batteries, and so it is configured to turn off when the rest of the system does or at some finite duration after activation.

IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

The present invention is generally directed to a system and method that facilitate the distribution of resources of devices to other devices. The present invention is ordinarily not involved with what the devices do with each other's resources, what is communicated at that time, or how the communication between devices takes place. Rather, the present invention can be generally considered more as a structured way for devices to find out about each other's resources, capabilities and other details, and then allow the devices to share resources on their own. Although not necessary to the present invention, as part of facilitating the resource distribution, security is provided for practical purposes, at least to ensure that valid credentials are required to use another device's resources. Thus, the present invention is preferably identity-based, in that distributed resources have relationships with one another according to identity data (e.g., credentials) provided to the devices.

In general, the present invention is directed to distributing computing among a user's devices, wherein a user (or owner / administrator) is anyone authorized to control and/or use the devices. However, because in general devices are increasingly becoming disaggregated, the methods and

systems described herein may be more properly described as distributed computing of resources. For example, a single distributed device may be made up of many resources, but may share as little as one its resources (e.g., only its non-volatile storage) to participate as a resource provide. Notwithstanding, as used herein, the terms "device" and "resource" are generally interchangeable except where otherwise apparent, e.g., using another device or using a resource of the other device to accomplish some computing task are generally considered the same thing. Also, as used herein, bandwidth is a device resource, even though bandwidth may not ordinarily be considered as being associated with any particular device.

As generally represented in FIG. 2, a distributed computing environment 200 constructed in accordance with various aspects of the present invention includes a first device 202 (Device A, such as the computer system 110 of FIG. 1A) and a second device 212 (Device B, such as the computer system 188 of FIG. 1B), along with possibly other devices (not shown). Each device that participates in the distributed computing environment 200 includes a distributed computing manager, 204 and 214. In general, and as described below, the distributed computing managers 204 and 214 may comprise some amount of code that communicate with one another over some

medium, e.g., wireless FM, line-of-sight such as infrared, satellite, a direct connection, the Internet, or an intranet. Indeed, more than one such medium may be available, and the distributed computing managers 204 and 214 may choose which
5 one or ones are best for a given circumstance.

An initial purpose of the distributed computing managers 204 and 214 are to locate each other. Such location may be automatic, such as by having each device occasionally broadcast its availability to others and/or by polling for the
10 availability of other devices, or alternatively may operate in response to some specific request from a program (or user) to connect to another device. As represented in FIG. 2, the distributed computing managers 204 and 214 each include a security mechanism 206 and 216, respectively, that exchange
15 credentials 207, 217 or the like (e.g., obtained from an authentication service 220) to ensure that the user who is currently operating the device which is contacting it (and/or the device itself), is valid in the distributed computing environment, and similarly, if necessary, that the device
20 which is being contacted is valid. Note that a program being used may also factor into the security / access control, e.g., a first program running on one device may have authority to use another device's resource or resources, but not another program, independent of or in addition to the user identity

and/or the device identity. Further note that the security mechanism may operate per identity and per resource, e.g., one entity may be authorized to use a device's display, but not its storage, while another entity may be able to use any of that same device's available resources, including its storage. This is similar to the .NET security model in which services provide methods that control access to a users' data based on the requesting user's identity, defined role and scope for that role. In this way, data can be accessed by its owner, and shared to an extent determined by the owner. In the present invention, instead of data it is a device resource being accessed, whereby a user's identification, defined role and scope for that role determine the extent to which a resource may be accessed.

Indeed, the security mechanisms of the present invention are based on those essentially those employed for security / access control in .NET services, as generally described in United States patent applications, serial nos. 10/003,754 entitled "IDENTITY-CENTRIC DATA ACCESS" and serial number 10/003,767, entitled "AUTHORIZING A REQUESTING ENTITY TO OPERATE UPON DATA STRUCTURES" filed on October 22, 2001, assigned to the assignee of the present application, ~~and herein incorporated by reference.~~ To this end, at least a thin subset of the .NET service runs on each device which

expects to advertise and share computing resources in the environment 200. In addition, a localized internal consistent security model may be executed on these devices such that local security is provided even when two local peer-to-peer devices may be disconnected from the centralized (e.g., Passport) authentication service 220 (FIG. 2).

The present invention is not limited to any particular transmission medium, but provides that the distributed resource services may be directly addressable to the target device, or may be queried via the Internet or an intranet, such as via the centralized set of .NET services on the web. When connected via the Internet or intranet, a program may query the last known computing resources available on any device associated with the user's identity, even when the device is offline or otherwise inaccessible. Bandwidth is also a resource, and may impact how a device's resources are shared, e.g., it may be more efficient to leverage a personal computer's resources when the personal computer has a high-speed connection to the Internet, but not if the personal computer connects only at low speed, such as if the other device has a high-speed connection. In fact, the personal computer may want to leverage the high-speed connection of the other device, in which event that other device may then leverage a resource of the personal computer, e.g., the

devices exchange resources to provide a more-optimal user experience.

In accordance with one aspect of the present invention, the distributed computing managers 204 and 214 also each
5 include a respective discovery mechanism, 208 and 218, that allows the devices to discover another participating device's shared resources / capabilities, that is, when that other device is configured to share its resources. The respective resource data 209 and 219 is formatted (e.g., in XML) and
10 exchanged in accordance with a defined schema, as described below. As represented in FIG. 2 by the exchange of the schematized data 209 and 219, both devices 202 and 204 are configured to share their resources ($R_{A1}-R_{Am}$ and $R_{B1}-R_{Bn}$) with other devices in the distributed computing environment.
15 However as can be appreciated, in a given environment, a particular device (e.g., a mobile telephone) may be only a consumer of resources rather a provider of any resource, or conversely may be only a provider, and not a consumer.

A device that is sharing its resources need not directly
20 communicate the resource data with the other device, but instead, for example, can advertise its resources somewhere for other devices to discover. For example, a .NET service (.NET Devices) is provided that centrally stores device resource capabilities such that data may be appropriately

536

formatted before sending it to a device, e.g., so as to not
send high-resolution graphics to a device incapable of
displaying them. A similar .NET Resources service, or an
extended .NET Devices service, may be provided to advertise
5 this information to other devices.

The following tables set forth some example schemas that
may be used (e.g., with XML-formatted data) to exchange
resource (storage and processing power) information regarding
a user's devices, which in this example include work and home
10 personal computers, and a cellular telephone:

```
<myStorage>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
</myStorage>
```

212

```

<myProcessingPower>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
</myProcessingPower>

```

As is understood, the schema is preferably resource based, i.e., multiple devices may be listed for each resource type, to match the general concept of an identity having distributed resources available for use, independent of the actual device on which the resource exists. Notwithstanding, the present invention is not limited to any particular schema and/or formats, e.g., the schemas may be device-based, such that each device may instead provide its list of available resources to other devices such that the same information is known.

As generally represented in FIG. 3, an architecture 300 is provided in order for devices to participate in an intelligent distributed computing model, in which various

aspects of the participating devices are abstracted and made programmatically addressable from a different device. A universal plug-and-play (UPnP 2.0) infrastructure or the like may be used as the basic, low-level mechanism 302 for
5 discovery of these resources. Alternatively, the list of resources for various computing devices may be cached in some well-known location, such as an Active Directory store, to enable querying of remote resources even if the device is only accessible through some intermediary such as a firewall or
10 wireless hub. Atop the discovery mechanism are the schemas 304, e.g., one schema associated with each specific type of the computing resource, with each of those schemas exposed, for example, as in the .NET architecture.

Note that while UPnP provides a suitable discovery
15 mechanism 304, it is only part of the process of actually using the remote computing resources. Thus, UPnP and .NET technologies are used to facilitate resource sharing, but are not used to actually employ the resources available on a device. Rather, once the resources are known, appropriately
20 focused protocols for that specific resource domain are used. By way of example, UPnP, with the addition of the .NET identityHeader, endpointHeader, and requestHeader SOAP semantics, is well-suited to obtain the video screen characteristics of the remote device, however a protocol such

as RDP will be used to transfer video data after the initial discovery and handshaking occurs between the devices. Note that this is similar to the way .NET approaches data services today, e.g., .NET provides a mechanism to connect to an electronic endpoint representing a proprietary (e.g., Messenger) protocol, but once that endpoint is connected, .NET allows the client to use an existing real-time protocol. Other protocols appropriate for other types of resources are used, e.g., CIFS+ for storage, which scales well to both corporate and home storage scenarios and enables the user's file system to continuously expand to other storage devices, without forcing the user to deal with drive letters, namespaces, and so forth. In summary, the present invention provides an identity-based approach to discovering these additional resources that are available for use, but does not thereafter directly deal with their use.

FIG. 6 is an example flowchart that represents some general aspects of the present invention, beginning with step 600 which represents receiving a request to use a shared resource. The request may be general, e.g., requesting the list of resources that a user has available on devices configured to participate, or may be a request for a specific resource, e.g., in response to an advertisement. For purposes

Gichilly

6. A computer-readable medium having computer-executable instructions to perform ~~the~~ ^{of one of claims 1 to 5.} a method for ~~facilitating accessing the computing resources of devices by other~~ devices, the method comprising:

at a first computing device (202, 400, 500) associated with at least one identity;

locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;

selecting a resource of a second device from the information, the second device being associated with the at least one identity;

accessing the resource of the second device; and

performing a computing task including accessing the resource of the second device ~~to perform at least part of the computing task.~~

allowing a mobile computing device not having a printer driver to print on a printer, facilitating accessing the computing

7. A system in a computing environment for ~~resources of devices by other devices~~, the system comprising:

~~a first~~ ^{having a printer driver} computing device (212, 404, ~~504~~) associated with at least one identity, the first device ~~including a resource (R_{B1} - R_{Bn})~~ and configured to provide access to the ~~resource by at least one other device (202, 400, 500)~~ ^{printer driver} associated with the identity, ~~the first computing device providing~~ information (209B) about the ~~resource~~ ^{printer driver} ~~according to a resource schema associated with the at least one identity;~~

~~a second device (202, 400, 500), the second device~~ ^{being a mobile computing device} ~~configured to request access~~ ^{to the printer driver} ~~to the resource; and~~ ^{to the printer driver}

the second device performing a ^{printing} ~~computing~~ task including accessing the ^{printer driver} ~~resource~~ on the first device to perform at least part of the ^{printing} ~~computing~~ task.

- ~~8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).~~

9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.

10. The system of claim 7 wherein the resource comprises processing cycles of a processor.

- ~~11. The system of claim 7 wherein the resource comprises an input method.~~

① to the second device

Eichler

- ~~12. The system of claim 7 wherein the resource comprises an output method.~~
- 8 13. The system of ~~one of~~ claims 7 ~~to 12~~ further comprising a third computing device (402, ~~502~~) connected to the first computing device, wherein the second device accesses the ~~resource~~ ^{printer driver} to perform an operation with the third computing device,
- ~~14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.~~ ^{the}
- ~~15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.~~
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first ~~computing device to the third computing device.~~
- ~~17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:~~
- ~~a first field for identifying a type of resource;~~
 - ~~a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;~~
 - ~~a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.~~
18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes ~~information corresponding to total size data of the storage resource.~~

C. Chilly

- ~~20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.~~
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
- ~~29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).~~

DRUCKEXEMPLAR

EPO - Munich

55

30 April 2003

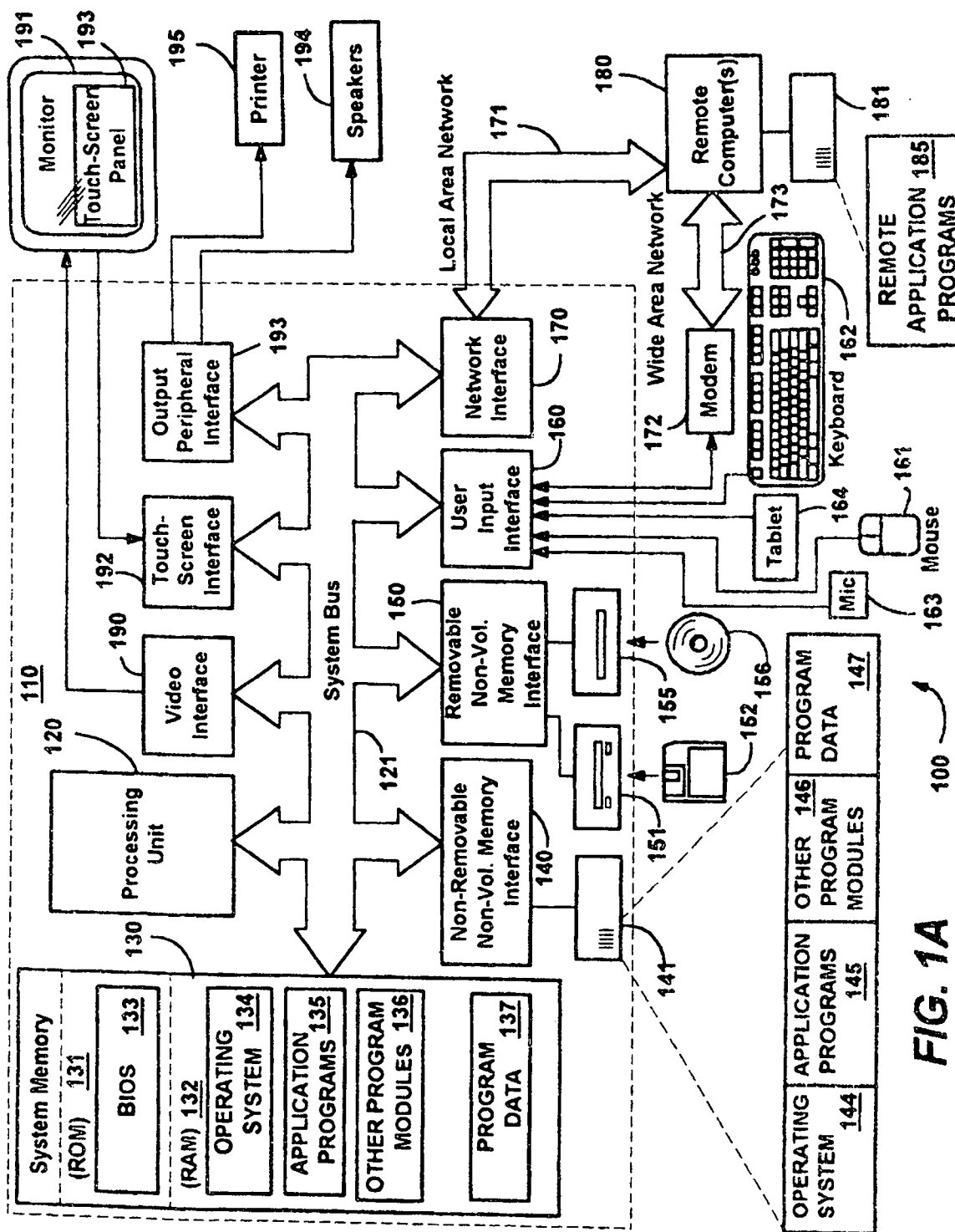


FIG. 1A

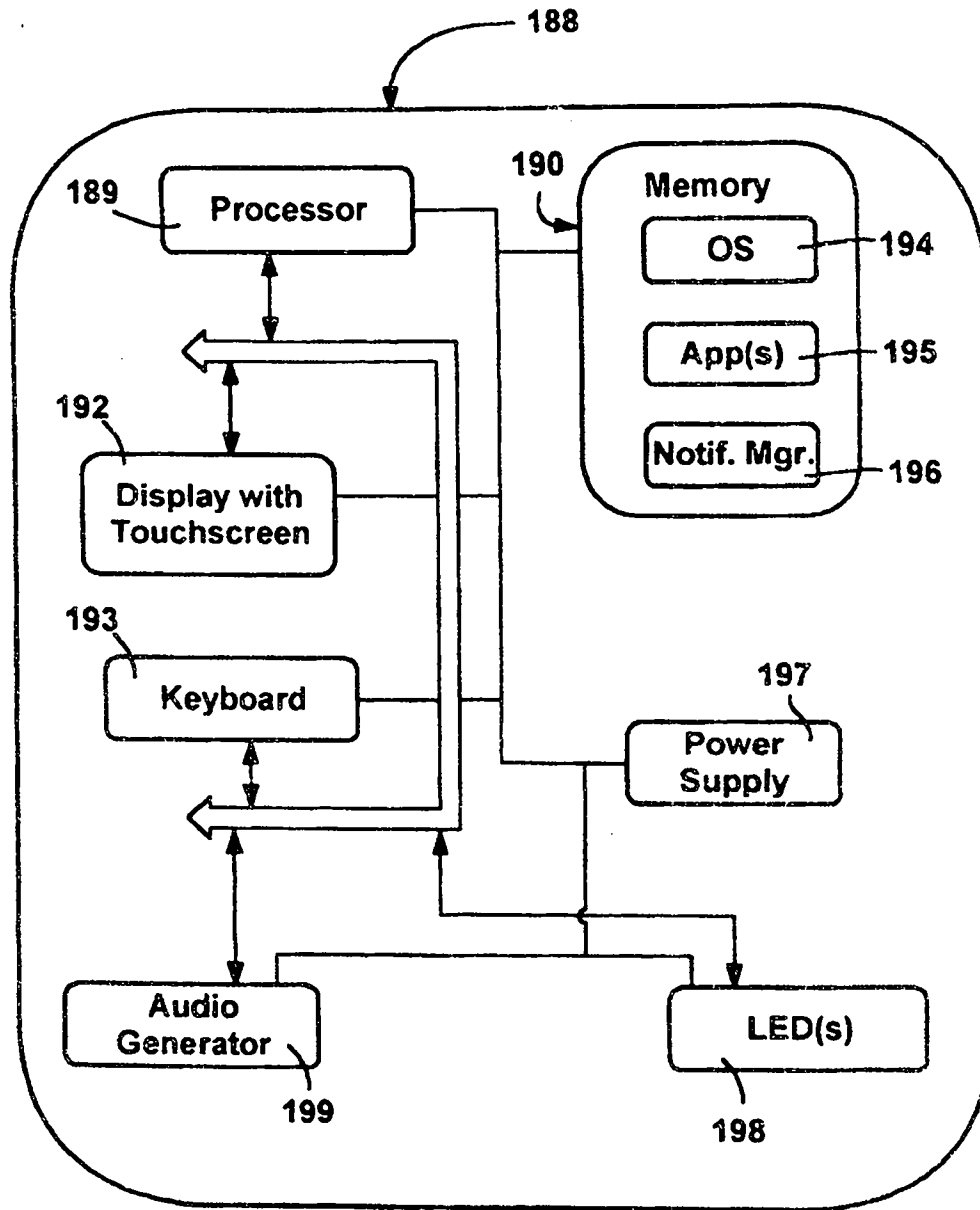
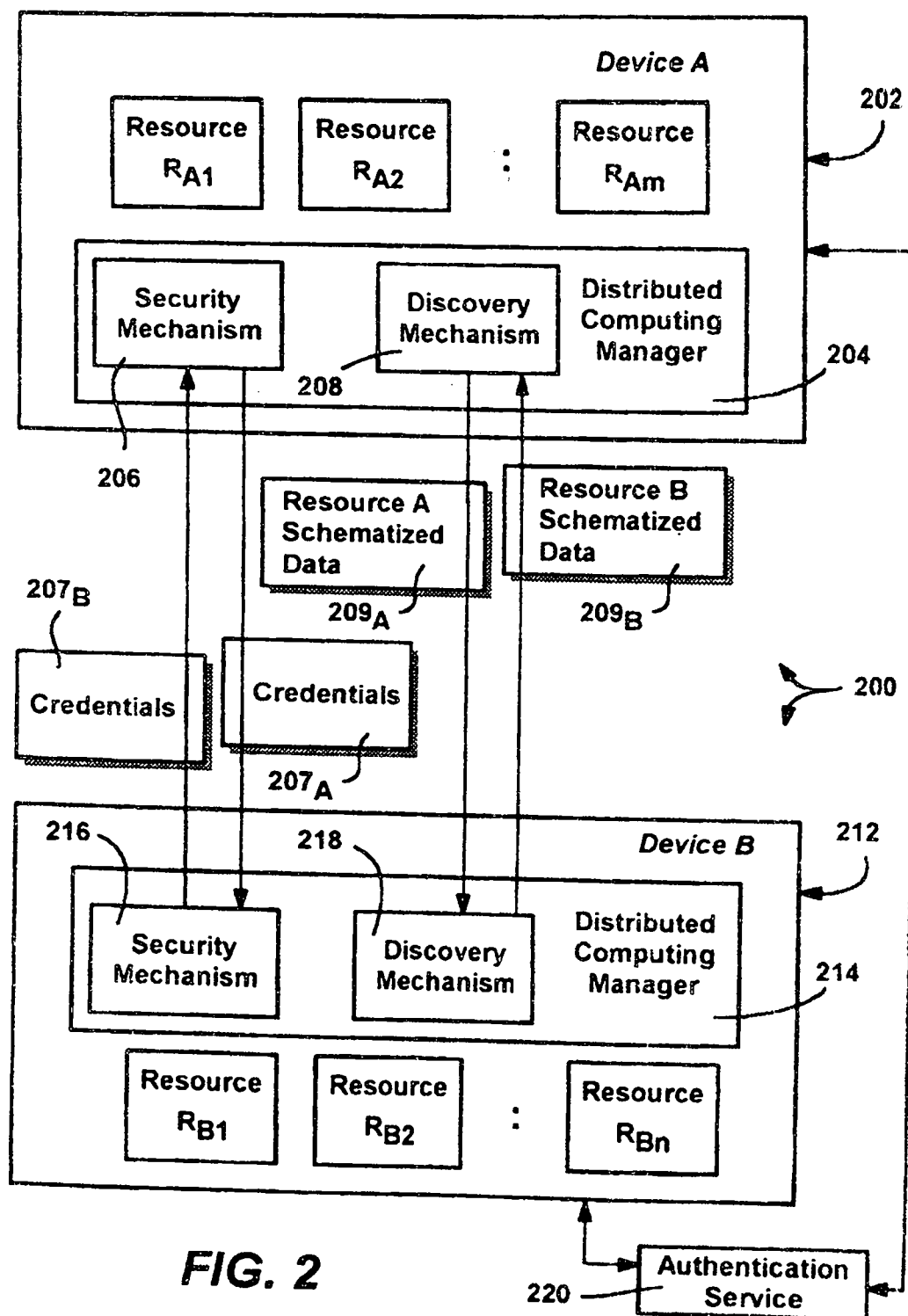


FIG. 1B



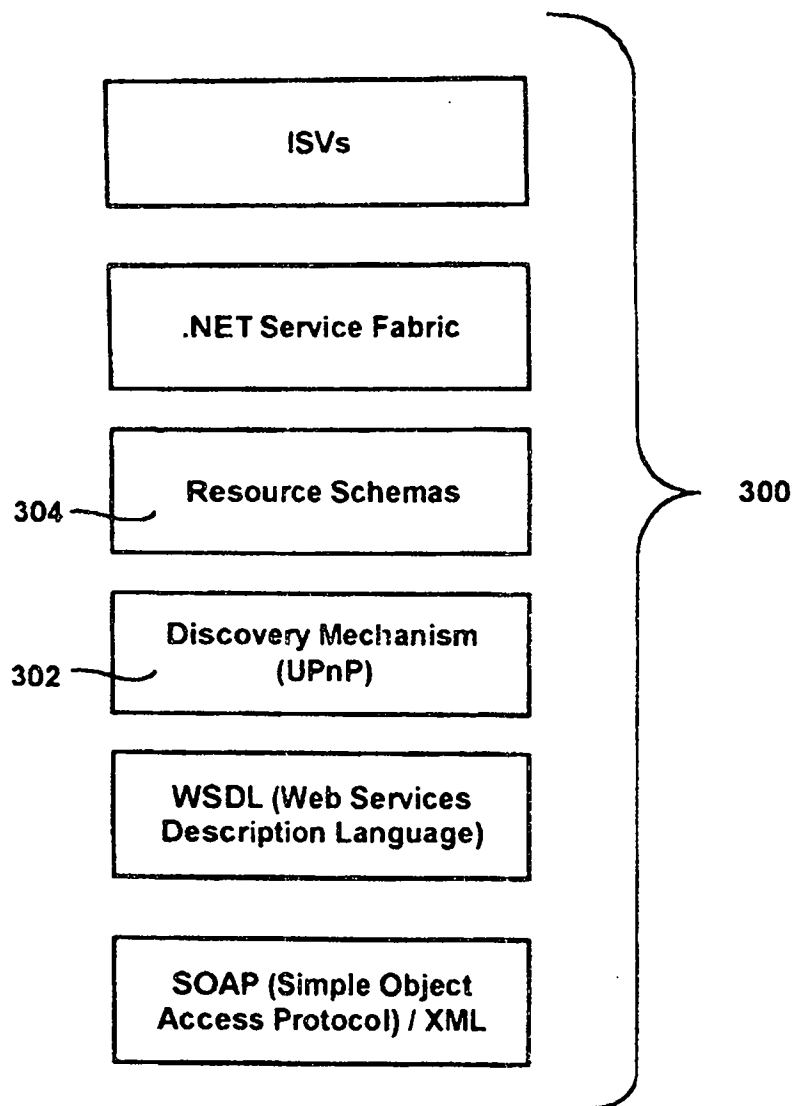


FIG. 3

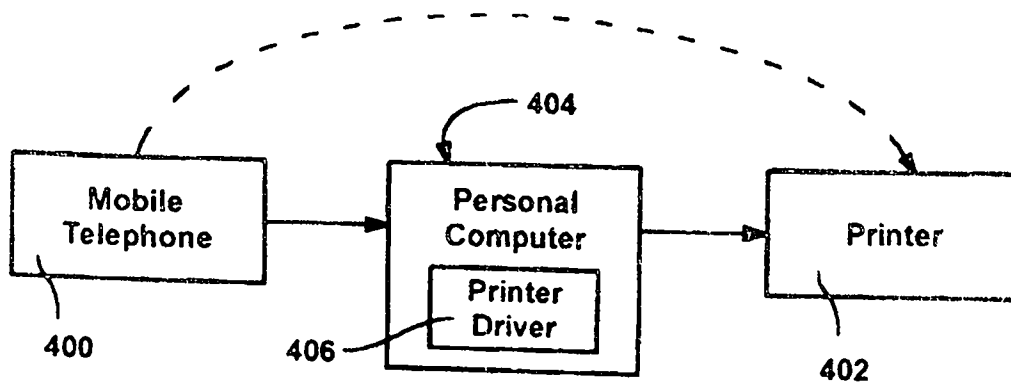


FIG. 4

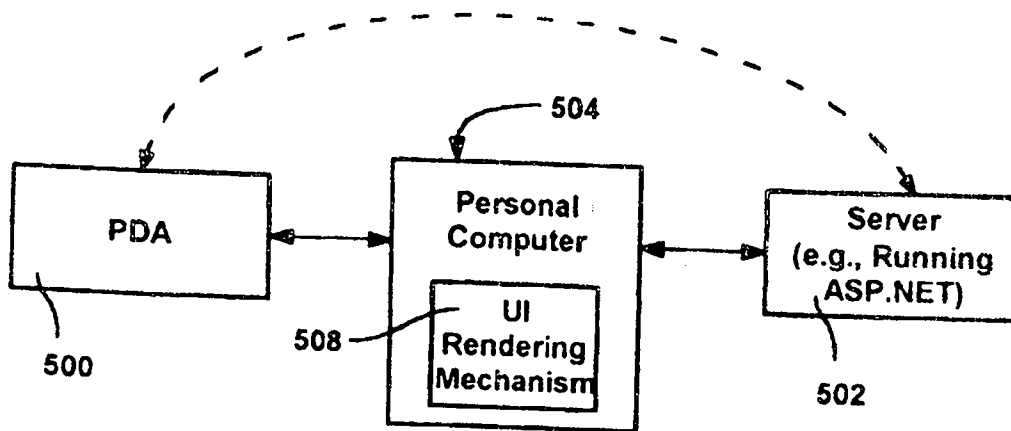
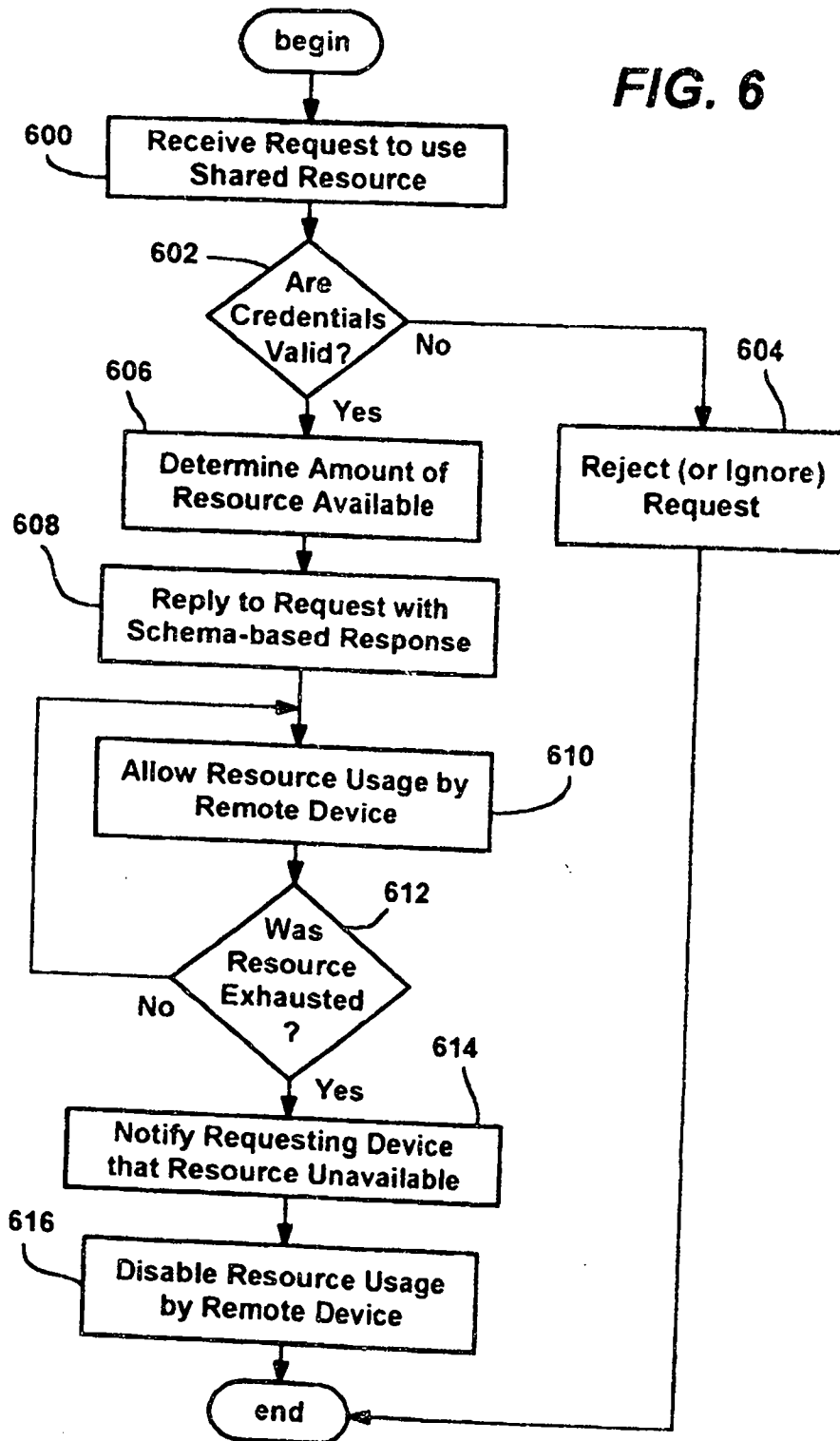


FIG. 5

FIG. 6



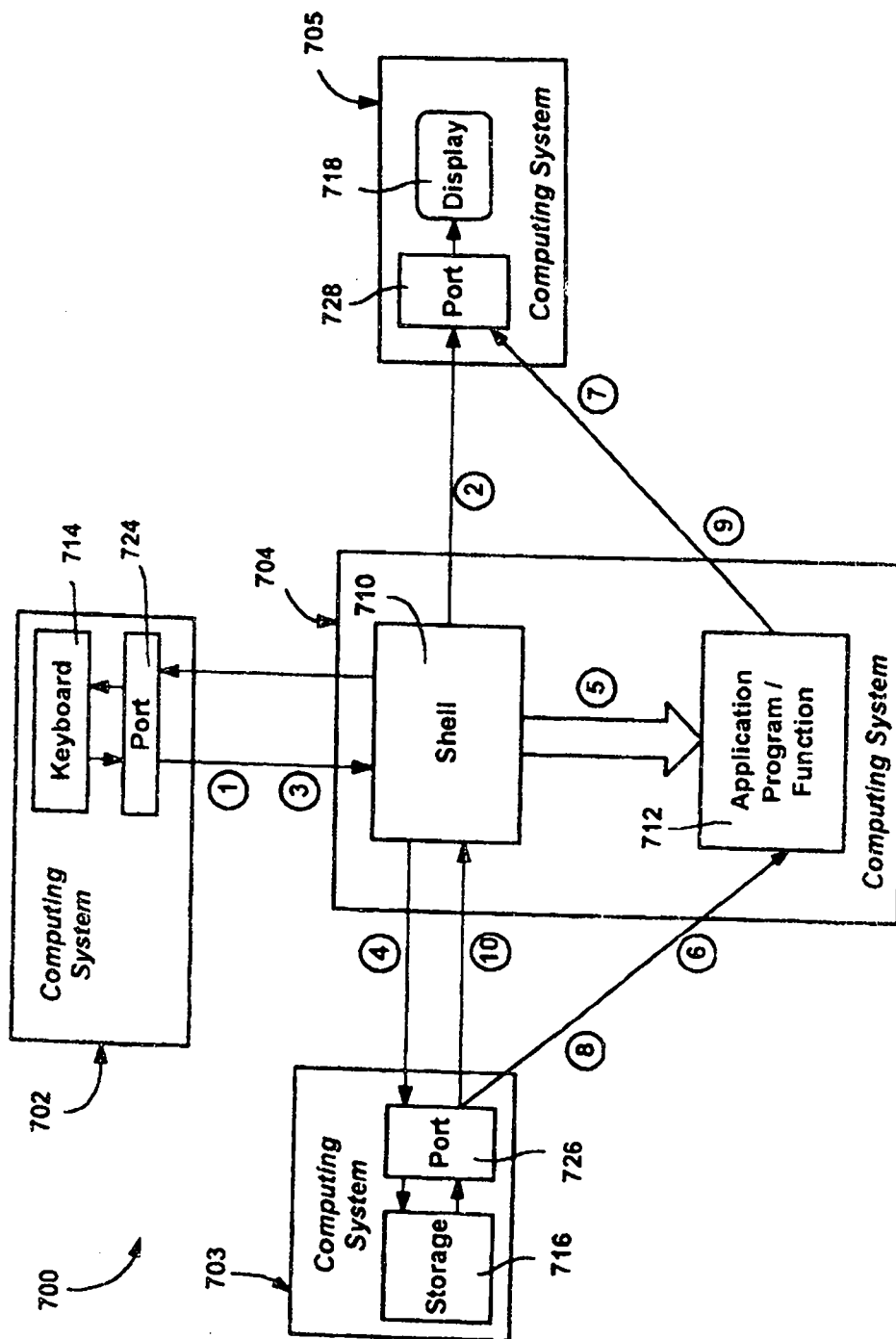


FIG. 7



EPA/EPO/EB
D-80298 München

+49 89 2399-0
TX 523 556 epmu d
FAX +49 89 2399-4455

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 2

Directorate General 2

Direction Générale 2

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Application No.
03 009 717.4 - 1244

Ref.
EP27314-034/kf

Date
28.02.2007

Applicant
MICROSOFT CORPORATION

Provision of a copy of the minutes in accordance with Rule 76(4) EPC

The attached copy of the minutes of the oral proceedings is sent to you in accordance with Rule 76(4) EPC.



Ragni, Marie-Astrid
For the Examining Division
Tel. No.: +31 70 340 - 3795

Branch at The Hague

Enclosure(s): Copy of the minutes (Form 2009)

Europäisches Patentamt
GD2

European Patent Office
DG2

Office européen des brevets
DG2

Application No. :

03 009 717.4

Minutes of the oral proceedings before the EXAMINING DIVISION

The proceedings were not public.

Proceedings opened on 15.02.2007 **at** 10:07 **hours**

Examining Division:

Chairman:	Kesting, Volker
1st member:	Goya, Jesus
2nd member:	Peeters, Dirk
Minute writer:	Peeters, Dirk

Present as/for the applicant/s:

Mr. Eickelkamp

accompanied by: Mr. Schornack

The identity of the person/s present and, where necessary, the authorisation to represent/authority to act were checked.

Essentials of the oral proceedings and the relevant statements of the applicant/s:



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 28.02.2007

Blatt
Sheet 1
Feuille

Annonce-Nr.
Application No. 03 009 717.4
Demande n°

The oral proceedings were **opened at 10:07**, after having checked the identity of the applicant's representative Mr. Eickelkamp through presentation of an attorney card (id. 98040) and the person accompanying, Mr. Schornack, through presentation of his identity card.

1 Introduction

The chairman described the state of the case, and indicated that the document US5,748,890 mentioned during the telephone conversation on 29.01.2007 is introduced into the procedure and will be referred to as document D7.

The representative acknowledged that he maintained the main request and the five auxiliary requests filed on 15.01.2007 in reply to the summons to attend oral proceedings.

The chairman proposed to discuss independent claims 1, 6 and 7 first, and independent claim 17 afterwards.

2 Main request

The representative pointed out the differences of claim 1 with respect to the closest prior art D1, the differing features being the feature of the identity, the feature of discovering, the feature of the first device performing the computing task with the second device only performing part of that computing task, and the feature of the first device accessing the resources on the second device.

The chairman maintained the argumentation that independent claims 1, 6 and 7 are novel, but that they lack an inventive step, as already mentioned in the minutes of the telephone conversation.

The first member continued the argumentation of the chairman, referring to the summons an to the telephone conversation, and pointed out the correspondence between the supposedly differing features of claim 1 and the features disclosed in D1, with only the feature of the identity lacking from D1, but being known to the person skilled in the art, for example being known from D7.



The representative asked the first member to indicate which devices from D1, in particular D1, figure 1, correspond to the first and second device of claim 1.

A break requested by the examining division was taken at **10:35**.

The oral proceedings resumed at **10:45**.

The first member referred to the first official communication of 02.09.2004, and pointed out in detail the correspondence between the devices disclosed in D1 and the first and second device of claim 1, the first device corresponding to computer 2 and the second device corresponding to computer 3, for example.

The representative asked why the skilled person would combine D1 with D7.

The first member argued that the features of D7 are features commonly known to the person skilled in the art, and that the skilled person, when confronted with the problem of how to secure the access to resources, would find in the state of the art documents pointing at this problem and its solution, with D7 being an example document disclosing such problem and the same solutions.

A break requested by the representative was taken at **11:10**.

The oral proceedings resumed at **11:20**.

The representative stated they changed their view with respect to document D1, and asked the division why the auxiliary requests are not allowable.

The chairman pointed out, before treating the auxiliary requests, that the feature of identity in claim 1 was very broad and not even claimed to be used to restrict access, and that D7 was not even needed to refute the inventiveness of claim 1, and re-iterated that independent claims 1, 6 and 7 lack an inventive step.

3 First Auxiliary request

The representative presented and explained his first auxiliary request, with the important differing feature being that the at least on identity identifies a user.



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum:
Date 28.02.2007
Date

Blatt
Sheet 3
Feuille

Annonce-Nr.
Application No. 03 009 717.4
Demande-Nr.

The first examiner stated that claim 1 lacks an inventive step, based on the disclosure of D1 and the common knowledge of the skilled person as exemplified by D7. D1 discloses a system being accessed over a WAN by users using terminals (D1, column 6, lines 8-14), in such a situation it is obvious to control access of users and terminals to the system, for example by using the features disclosed in D7.

The representative argued that D7 is a patent and therefore is not common knowledge but an invention in its own respect, and therefore would not be combined with D1.

The chairman pointed out that the feature of the identity in claim 1 was still very broad and not even claimed to be used to restrict access, and that D7 was not even needed to refute the inventiveness of claim 1, and that even if claim 1 were drafted such that the identity would be used to restrict access, this differing feature would be known from D7, with D7 being an example document, parts of which being common knowledge.

4 Second Auxiliary Request

The chairman continued with the second auxiliary request, with the differing feature being that the resource rather than the device is associated with the identity, and stated that in the description resource and device can be used interchangeably, and that therefore the argumentation for the first auxiliary request applies also to the second auxiliary request.

5 Third Auxiliary Request

The chairman continued with the third auxiliary request, with the differing feature being that the user's identification determines the extent to which the resource may be accessed, and that allowing or restricting access is also an extent, the extent being 0 or 1 but nonetheless an extent.

The applicant asked that if a new claim would be drafted with the extent being defined as a range between 0 and 1, if such a claim would be considered inventive.

The first examiner argued that an account indirectly determines an extent, and that the assessment of inventive step would depend on the exact wording of the claim, but that in



any case the feature of the extent is part of the common knowledge.

6 Fourth Auxiliary Request

The chairman continued with the fourth auxiliary request, with the differing feature being that only the computing task performed by the first computing device is authorized to access the resource of the second device, and stated that the expression "only" is not clear.

The first examiner continued to explain why the expression "only" was not clear, as it can relate to computing tasks on the first device or to computing tasks on the second device, and that even if it were clear it would be considered a feature commonly known.

In reaction to this, the representative referred to the description, at the bottom of page 19, further describing how "only" should be interpreted.

A break was taken at **11:45**.

The oral proceedings resumed at **12:40**.

The chairman stated that the differing feature of claim 1 is an obvious feature, and substantiated this by giving the example of the resource being memory, whereby different computing tasks are obviously not allowed to access one another's memory because otherwise the computing tasks would obviously interfere, and with the example of the resource being program modules, and that likewise computing tasks are obviously not allowed to access one another's program modules because otherwise the computing tasks would obviously interfere.

Upon being asked to comment on this, the representative stated that his invention is different, referring to the description page 4 lines 15-18 and page 19, but stated that this is not claimed.

7 Fifth Auxiliary Request

The chairman continued with the fifth auxiliary request, with the differing feature being that data including resource information of the second device is formatted according to a



schema understood by the first device. The chairman stated that this feature is already known from D1, figure 2.

The representative agreed to have a break, during which he would draft a new claim 1.

A break was taken at **13:00**.

The oral proceedings resumed at **13:50**.

8 Draft Auxiliary Request

The representative presented his draft new claim 1, which is based on the original claim 1 and the description page 30 lines 9-10 and line 23, page 31 lines 2-4, and figure 4, and which is now more remote from the disclosure of D1.

The chairman asked the representative to explain why D1 is not anymore the closest prior art. The representative replied by stating that the new claim 1 defines a method for a mobile device to perform a print job, whereas D1 discloses a load balancing method in a parallel computer system.

The chairman asked the representative to provide arguments why the new claim 1 contained an inventive step. The representative provided some arguments based on a hypothetical prior art contained in the description pages 30-31.

A break was taken at **14:00**.

The oral proceedings resumed at **14:55**.

In response to the newly drafted claim 1, the chairman stated that D5 now represents the closest prior art, and that the following features contravene Art. 123(2) EPC: the device defined in claim 1 is a mobile device, whereas the application always refers to computing devices, and claim 1 does not contain the feature of the identity, whereas in the application this feature is consistently present in both the embodiment of the description on pages 30-31 and in the corresponding claim 14 as published.

The first examiner continued by pointing out the correspondence between the features of



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 28.02.2007

Blatt
Sheet 6
Feuille

Annonce-Nr.
Application No. 03 009 717.4
Demande n°

D5 and claim 1, the only differing features now being that the device of claim 1 is a mobile device, and that the resource information of claim 1 includes printer driver information.

The chairman asked the representative to provide argument in favour of the inventiveness of claim 1.

The representative stated that in the disclosure of D5, there is no need for the mobile device to know anything about printer drivers, and that there is no motivation in D5 to modify the resource information to include printer driver information. The differing feature of claim 1 allows the mobile device to decide on the quality of the printing task by selecting a printer driver capable of handling particular file formats.

The representative was allowed a break to modify new claim 1 so as to restore it's compliance with Art. 123(2) EPC, and to redraft independent claim 6 and 7 and the dependent claims.

A break was taken at **15:15**.

The oral proceedings resumed at **15:35**.

The representative stated that he maintains the Main Request and that the set of newly drafted claims 1 to 8 now replace all the former Auxiliary Requests, the set of newly drafted claims thereby becoming the sole Auxiliary Request.

The first examiner commented on independent claim 17 of the Main Request, this claim 17 also being the same in all former First to Fifth Auxiliary Requests, and stated that independent claim 17 of the Main Request does not involve an inventive step with respect to the disclosure of D2, referring to the official communication of 12.09.2005, point 2.3. The representative did not wish to further comment on this.

A break was taken at **15:45**.

The oral proceedings resumed at **15:55**.

The chairman proposed changes to the text of claim 7 and 8, to solve some clarity



problems concerning claim 7:

- line 5: "one other device" change to "a second device"
- line 6: in line 6 it should read "the first computing device providing to the second device information"
- line 8: "a second device" change to "the second device"
- line 8: "a mobile computing device" change to "the mobile computing device"

concerning claim 8:

- line 4: in line 4 it should read "wherein the third computing device comprises the printer".

The chairman additionally proposed that the examining division would adapt the description to the newly filed claims and would include a reference to the prior art in the description.

The representative agreed to the changes to the text of the claims, and to the proposed changes to the description.

The chairman announced the following **opinion** of the examining division:

- The Main Request does not meet the requirements of Art. 56 EPC in view of document D1 and the common knowledge of the skilled person concerning claims 1, 6 and 7, and in view of D2 and the common knowledge of the skilled person concerning claim 17.
- The Auxiliary Request including the abovementioned amendments meets the requirements of the EPC.

9 Conclusion

The chairman announced the intention of the examining division to grant the First Auxiliary Request filed during today's oral proceedings and including the abovementioned amendments and to issue a Rule 51(4) communication.

The oral proceedings ended at **16:00**.

**Bescheid:Protokoll (Anlage)**

Datum
Date 28.02.2007
Date

Communication/Minutes (Annex)

Blatt
Sheet 8
Feuille

Notification/Procès-verbal (Annexe)

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

D. Peeters
Second Examiner

The chairman informed the applicant/s as follows:

The examining division intends to grant a European patent based on the following application documents:

Auxiliary Request 1

Description, Pages

1-3, 5-19, 21-36, 38, 39	as originally filed
4, 4b, 20, 37, 40	received on 10.01.2005 with letter of 05.01.2005
4a	received on 22.03.2006 with letter of 22.03.2006

Claims, Numbers

1-8	filed during Oral proceedings on 15.02.2007
-----	---

Drawings, Sheets

1/7-7/7	as originally filed
---------	---------------------

With the following amendments to the above-mentioned documents by the examining division

Description, Pages	41)
Claims, Numbers	7-82)

Comments

1) Rule 27(1)(b) (acknowledgement of prior art document).
 2) Article 84 EPC, text as agreed with the representative during the oral proceedings.
 The examining procedure will be continued by issuing a communication under Rule 51(4) EPC on the basis of the above documents.

The chairman closed the proceedings on 15.02.2007 at 16.00 hours.

signed:

Kesting, Volker

Chairman

Enclosure(s):
 Auxiliary Request



signed:

Peeters, Dirk

Minute Writer

15.02.07

Eichlly

Application No.: 03 009 717.4 - 1244

Applicant: MICROSOFT CORPORATION

Our Ref.: EP27314TE109

Date: January 15, 2007

AUXILIARY

MAIN REQUEST

computing
mobile device printingof allowing a mobile device ^{<->} not having a printer driver to print on a printer,

1. A method in a computing environment for ~~facilitating accessing the computing resources of devices by other devices~~, the method comprising:
 - ~~at a first computing device (202, 400, 500) associated with at least one identity,~~
^{the mobile device} discovering ~~a second computing device (212, 402, 404, 502, 504) associated with the at least one identity,~~
^{another} ~~equipped with a printer driver (406) for said printer;~~
^{the mobile device} ~~at the first device,~~ obtaining data (209B) related to the ~~second~~ ^{another computing} device, the data including ~~resource~~ ^{printer driver} information of the ~~second~~ ^{another computing} device formatted according to a schema understood by the ~~first~~ ^{mobile <->} device;
 - ~~the first device interpreting the data to access a resource (R_B, R_{BN}) of the second device; and~~
^{mobile <->} ~~the first device performing a computing task including~~ ^{the printer driver} ~~accessing the resource of the another second device to perform at least part of the computing task.~~ ^{providing printing data via}
2. The method of claim 1 further comprising:
 - ~~communicating (600) a request to use the resource from the first device to the another second device.~~
^{mobile <->} ~~computing~~ ^{mobile <->} ~~second~~ ^{another computing} device.
3. The method of claim 1 or 2 further comprising:
 - ~~communicating identity information from the first device to the second device.~~
^{mobile <->} ~~first~~ ^{another computing} ~~device.~~ ^{second}
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

<computing>

Cichully

- 2 -

6. A computer-readable medium having computer-executable instructions to perform ~~the~~ ^{of one of claims 1 to 5.} a method for ~~facilitating accessing the computing resources of devices by other~~ devices, the method comprising:

at a first computing device (202, 400, 500) associated with at least one identity;

locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;

selecting a resource of a second device from the information, the second device being associated with the at least one identity;

accessing the resource of the second device; and

performing a computing task including accessing the resource of the second device ~~to perform at least part of the computing task.~~

7. A system in a computing environment for ~~facilitating accessing the computing resources of devices by other devices~~, the system comprising:

~~a first computing device (212, 404, 504)~~ ^{allowing a mobile computing device not having a printer driver to print on a printer,} associated with at least one identity, the first device ~~including a resource (R_{B1} - R_{Bn})~~ ^{having a printer driver} and configured to provide access to the ~~resource~~ ^{printer driver} by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the ~~resource~~ ^{printer driver} according to a resource schema associated with the at least one identity;

a second device (202, 400, 500), the second device ^{being a mobile computing device} configured to request access to the ~~resource~~ ^{printer driver}; and

the second device performing a ^{printing} ~~computing~~ task including accessing the ~~resource~~ ^{printer driver} on the first device to perform at least part of the ~~computing~~ ^{printing} task.

- ~~8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).~~

9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.

10. The system of claim 7 wherein the resource comprises processing cycles of a processor.

- ~~11. The system of claim 7 wherein the resource comprises an input method.~~

Lichulky

- ~~12. The system of claim 7 wherein the resource comprises an output method.~~
13. The system of ~~one of claims 7 to 12~~ further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the ^{printer driver} resource to perform an operation with the third computing device,
- ~~14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.~~
- ~~15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.~~
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first ~~computing device to the third computing device.~~
- ~~17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:~~
- ~~a first field for identifying a type of resource;~~
 - ~~a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;~~
 - ~~a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.~~
18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes ~~information corresponding to total size data of the storage resource.~~

Lidell

- ~~20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.~~
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
- ~~29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).~~

The chairman informed the applicants as follows:

The examining division intends to grant a European patent based on the following application documents:

Auxiliary Request 1

Description, Pages

1-3, 5-19, 21-36, 38, 39	as originally filed
4, 4b, 20, 37, 40	received on 10.01.2005 with letter of 05.01.2005
4a	received on 22.03.2006 with letter of 22.03.2006

Claims, Numbers

1-8	filed during Oral proceedings on 15.02.2007
-----	---

Drawings, Sheets

1/7-7/7	as originally filed
---------	---------------------

With the following amendments to the above-mentioned documents by the examining division

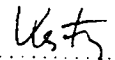
Description, Pages	41)
Claims, Numbers	7-82)

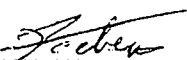
Comments

- 1) Rule 27(1)(b) (acknowledgement of prior art document).
- 2) Article 84 EPC, text as agreed with the representative during the oral proceedings.

The examining procedure will be continued by issuing a communication under Rule 51(4) EPC on the basis of the above documents.

The chairman closed the proceedings on 15.02.2007 at 16:00 hours.


Kesting, Volker
Chairman


Peeters, Dirk
Minute Writer

Annex(es):
Auxiliary Request

of this example, the request is treated as a request for one specific resource.

Step 602 represents the evaluation of the credentials, such as provided by the Microsoft .NET Passport authentication service. The request may include the credentials, or the credentials may be separately exchanged, e.g., before, as part of, or after step 600. If not valid at step 602, step 602 branches to step 604 which represents rejecting the request. Note that depending on a given environment, the request without proper credentials may be ignored rather than specifically rejected.

If valid, step 606 is executed, and represents determining the amount of the requested resource that is available. For example, a resource such as storage may be limited because it has only limited free space, and further, a device may be configured to make only part of its storage space available for sharing, reserving the rest for itself. A resource such as processor cycles may be made available only to the extent the processor is not being otherwise used, and even then may be further limited so that another device can only take up to a maximum amount of the unused cycles. Any resource may have a time-out value associated with it, which may be further contingent upon its use, e.g., storage can be shared for no more than one hour per request, and the request

will be terminated if no file system requests are detected at least every two minutes. In this manner, a device may be configured to share its resources to an extent controlled by its owner / administrator.

5 Note that the resource access may be dynamic, e.g., unused processor cycles may be shared, however CPU usage is a number that fluctuates, and the requesting guest device may be prioritized along with other devices and/or processes on the device that is sharing the CPU. For example, guest users may
10 only consume a controllable percentage of resources when the owner of the device is not otherwise using them, such as bandwidth or processing power. Both CPU and bandwidth are examples of resources which are scheduled according to time constraints (e.g., cycles or bits per second) rather than non-
15 time-based resources such as disk or memory usage. As a result, the scheduling mechanisms for time-based resources in general will be fundamentally different than for non-time-based resources. Note that kiosk locations would enable many scenarios that would be useful to such guests. A resource
20 such as a display monitor may be either entirely available for sharing or not at all, depending on whether or not it is presently being used. Thus, although not represented in FIG. 6, step 606 may also reject (or ignore, if appropriate) a request when there is no amount of a requested resource

available for sharing. Step 608 represents returning a response to the requesting entity, preferably the schema-based response that provides the resource capability data to the requesting entity.

5 Step 610 represents allowing the requested resource to be shared, and includes, for example, the exchange of data between the requesting device and the sharing device so that the sharing performs some corresponding task for the requesting device. Step 612 tests whether the resource was
10 exhausted, e.g., timed out or is no longer available. Note that this is shown as a loop in FIG. 6 for purposes of simplicity, however it is readily understood that the actions may be event driven. Further, a step (such as step 606) may be included in the loop / used to fire the event, e.g., the
15 resource usage is dynamically monitored to determine when it is exhausted for the particular request. Note that a resource may be directly exhausted when the requesting entity informs the device that the resource is no longer needed.

Step 614 represents notifying the requesting device that
20 the resource is no longer available for its use. Step 616 represents disabling the resource usage, so that, for example, once exhausted the requesting device will need to make a new request if it wants to share resources.

FIGS. 4 and 5 provide examples of some of the many possible uses of the present invention. As will be understood from the examples of FIGS. 4 and 5, integrating computational proxy support into a network edge device or appliance may
5 result in a better user experience for users of an internal network.

In a first example represented in FIG. 4, a small mobile device 400 (such as a mobile telephone) wants to print to a printer 402, however the small device 400 does not have an
10 appropriate printer driver. This may be because the mobile device 400 does not have sufficient memory to install and maintain a printer driver needed for that printer 402, and/or does not have an easy way for the needed printer driver to be installed, and/or for other reasons. Note that this is a real
15 scenario, and one proposed solution is to emit XHTML-PRINT as a simplified PDL (Page Description Language) from the mobile device 400 to the printer 402, e.g., over some network connection (shown via the dashed line in FIG. 4). Although XHTML-PRINT is adequate to an extent, this solution has a
20 number of drawbacks, e.g., XHTML-PRINT is not WYSIWYG, cannot control line breaks, and generally provides a poor user experience.

However, if via the present invention, the mobile device 400 first dynamically discovers a personal computer 404

equipped with a proper printer driver 406, (and which the mobile device user is authorized to use), the mobile device 400 instead provides printing data via the personal computer 404 to the printer driver 406, which in turn will create a WYSIWYG output using PCL, Postscript or rasterization. As mentioned above, the present invention provides the discovery mechanism, and thereafter the mobile device 400 may communicate in any way with the personal computer 404, such as by a lightweight PDL that has much higher-fidelity (relative to XHTML-PRINT). The result is high-quality output rendered very quickly, by leveraging the faster processor and proper printer driver 406 in the personal computer 404.

FIG. 5 demonstrates a second example, generally directed to providing an improved user experience when operating a PDA (personal digital assistant) 500 that is to be executing managed code remotely on a server 502 (e.g., an ASP.NET server). One way to accomplish this is to have the PDA connect to the server to run the managed code, (e.g., by leveraging Mobile Controls support in ASP.NET), with the results transmitted as HTML, as indicated by the dashed line in FIG. 5. The server 502 dynamically scales the HTML content to the appropriate screen size of the device 500, without the ISV creating special-purpose code for each device 500. However, this requires many round trips with the server 500,

with the screen flashing for updates, and generally provides a degraded user experience.

An improved solution provided by the present invention is accomplished by having the PDA dynamically discover that a personal computer 504 associated with the user's identity can be a proxy for a number of computational resources, as generally represented in FIG. 5. In such an event, the PDA issues an HTML request for a remote page via the proxy personal computer 504. The personal computer 504 intercepts the request and starts a session with the remote server 502 to offload the UI rendering managed code, e.g., using WSDL as the protocol to the server 502. Typically, the bandwidth from the proxy personal computer 504 to the server 502 will be less than the bandwidth from the PDA device 500 to the proxy personal computer 504, whereby the smaller amount of data transferred over the slower link to and from the server 502 results in a more responsive user interface when displayed on the PDA 500, providing an improved user experience. Of course, the two bandwidths can be first evaluated to ensure that the proxy solution is the more-optimal one.

Further, note that if the PDA 500 was equipped with locally-running controls that the server was able to leverage, an even better experience may be provided. For example, ASP.NET automatically leverages any local controls (e.g., such

as a date picker control) running locally on a PDA running a Windows®-based operating system. This results in even less round trips to the server 502 and/or proxy 504, with less resultant screen updating / flashing.

5 As can be appreciated, there are numerous other examples that are enabled by having computing resources revolve around an identity, instead of around the particular devices those resources ship with. The resource schemas and service fabric are not limited to simply describing the resources, but also
10 include elements to assist those resources working together. This can be appreciated from the following two types of scenarios, namely a set of real-time scenarios in which the user is interacting with a remote computing device in real-time, and a set of scheduled or predicted scenarios. An
15 example of a predicted scenario may include a storage load balancing mechanism that occasionally runs on a home network, to make sure that one computing device's free storage space is not going to run out while another drive has plenty of room.

 Thus, a rich schema may provide data that goes beyond
20 describing the resources themselves, such as methods to schedule, at a particular time or at regular intervals, prioritization machinery, and data indicating how to deal with possible contention between real-time and predicted tasks, or multiple such tasks. The schema may provide for the concept

of cost, e.g., if two connections are available to the Internet, it is valuable to recognize that the user may be paying one price for unlimited access but another price for an amount of data transferred over a different mechanism, and
5 select the better option.

Replication/sharing by a logging system may be used to understand how users are interacting with their devices, and to achieve a consolidated view of how a user operates, rather than have only disparate views separated by device. This data
10 is appropriately known to many devices, and will provide an automatic way for such data to be consolidated. Prediction optimizations may be saved and exchanged based on monitored usage, e.g., if the system knows that a user manually downloads particular information to a hand-held computer every
15 Friday before leaving work, the system can start the download during device idle time so the "synchronization" appears instantaneous. End user options/settings/preferences is also valuable information for the shared environment, e.g., what is a desirable or even necessary feature for some users is
20 undesirable or even annoying to others. By maintaining this data, the end user has the power to be in control of their experience. Also, in a typical environment, there will be at least one rich client that uses the protocol/schema to perform system management, where "system" refers to the overall set of

resources associated with an identity. Algorithms and methods can use the schema for predicting such operations.

As is understood, there are many valuable scenarios enabled by through abstracting device resources and
5 aggregating them around an identity. A number of examples are provided, however the list should not be considered exhaustive or in any way limiting, but rather serves to illustrate some of the many benefits and advantages of the present invention. As examples mentioned above, a device with a small computer
10 screen may leverage an unused computer monitor connected to a personal computer for better display, use resources of a proxy to facilitate a better display on its own display (FIG. 5) or print to a printer even without printer driver, by connecting to a printer via a personal computer that does have the
15 correct driver (FIG. 4).

Other examples can be seen in a very low-powered device, such as a mobile telephone, performing complex computations by using the memory and CPU of a personal computer. Stored data can be automatically and transparently moved from a first hard
20 drive to a second when the first hard drive is getting full, including, for example, from personal digital video recorders that record video programming and need to delete old shows to make room for more recent recordings. The present invention instead moves those older shows to another storage, rather

than delete them. A portable audio player need not be able to play any type of physical media, but can instead receive and buffer audio streamed to it from a personal computer and play the audio from memory.

5 Other examples include allowing users to easily access their files from any device regardless of where it is stored; for example, while on the road with only a small personal computer, being able to easily open files stored on a home personal computer or work personal computer, (independent of
10 centralized .NET data storage). A user may see all of the free space in various devices, including in various public sharepoints on a corporate network. A large-screen TV connected to an appropriate device may be used to view photographs saved on a personal computer folder, run a
15 slideshow, watch home movies or movie trailers saved on a personal computer, and so on.

FIG. 7 provides a demonstration of an implementation 700 run using four interconnected computing systems, 702-705. The implementation 700 includes a command shell 710 and
20 application program 712 (or similar function code) on the computing system 704, a keyboard resource 714 on the computing system 702, a storage (e.g., disk) 716 on the computing system 703, and a display 718 on the computing system 705. Thus, in this example the resources are distributed such that there is

576

an input resource the on computer 702, storage resource on the computer 703, and an output resource on the computer 704. The computer 704 acts as a coordinating authority for the implementation 700. Although the resources are on different
5 computer systems, the resources are accessed in accordance with the present invention to essentially form a single functioning system.

The shell resource 710 calls the application 712, and also connects to the other devices' resources through ports
10 724, 726 and 728, conceptually represented in FIG. 7 as one port per resource, although a separate input and output port may be provided. Ports essentially comprise structured queues that can be written (output) to or read (input) from via a handle to that port. Ports are mobile, data is not lost, and
15 security can be associated with a port such that no one except the intended entity will see the data intended for it.

The shell 710 used in the implementation 700 is based on a programming language referred to as xSpresso, as described in United States provisional patent application serial number
20 60/379,864, filed on May 10, 2002, ~~and herein incorporated by reference in its entirety/~~ In general, xSpresso is directed to a programming language and runtime that brokers control and data flows among asynchronous system components. Among other things, xSpresso expresses the cooperative dimensions of

251

programs or services, including ordering and timing, that can be programmatically verified. The language may programmatically express the passing of an organizational scheme formed from a customizable tag-based language such as

5 XML.

In the example of FIG. 7, the numbered arrows labeled 1-4 and 6-10 represent an example of the various communications between the ports 724, 726 and 728 and the shell 710, to display the text data of a file in the storage 716 to the

10 display 718, e.g., similar to the well-known "type" function. In the example of FIG. 7, a user of the keyboard resource 714 initiates the operation by typing the word "type" on the keyboard 714 (arrow labeled one (1)). The shell 710 receives this via the port 724 and communicates data to the port 728

15 for display on the screen 718, (arrow labeled two (2)), e.g., to echo what the user has input and/or to prompt for a filename of a file that stores the text that the user wants to have displayed. In FIG. 7, the arrow labeled three (3) represents the user typing in the filename, such as

20 corresponding to an XML file, which may also be echoed (not separately labeled).

With the filename, the shell 710 then writes data (represented by arrow labeled four (4)) to the port 726 that corresponds to the storage (e.g., disk) resource 716, which

results in a read request to the storage's file system. In parallel, the shell 710 calls (the wide arrow labeled five (5)) the application program 712 to receive and output the read data. In response, the storage resource 716 begins
5 outputting data via port 726 to the application program 712, which in turn outputs it to the port 728 of the display resource 718, as represented by the arrows labeled six (6) and seven (7). The arrows labeled eight (8) and nine (9) represent the rest of the data being transferred in a similar
10 manner. The arrow labeled ten (10) represents the status (e.g., success) code being returned to the shell 710.

In keeping with the present invention, storage, input and output can be placed on different systems, with no change to the implementation. Legacy interoperability may be provided
15 via common language runtime (CLR) classes that represent common legacy components (e.g., the keyboard, console and storage). System dispersion is provided in a local and transparently distributed implementation, as is explicit handling of concurrency and message-passing.

20 As can be seen from the foregoing detailed description, there is provided a method and system for distributed computing of device resources based around the concept of identity. The method and system abstracts the resources of a computer from the individual devices in a manner that provides

Printed: 21-01-2005

DESCNEW

03009717

numerous benefits to a user and generally improves the user's computing experience. The method and system are schema-based, and are extensible and highly flexible.

While the invention is susceptible to various
5 modifications and alternative constructions, certain
illustrated embodiments thereof are shown in the drawings and
have been described above in detail. It should be understood,
however, that there is no intention to limit the invention to
the specific forms disclosed, but on the contrary, the
10 intention is to cover all modifications, alternative
constructions, and equivalents falling within the ~~spirit and~~
scope of the invention.

15.02.07

Eichlly

Application No.: ~~03-009-717.4-1244~~

Applicant: MICROSOFT CORPORATION

Our Ref.: EP27314TE109

Date: ~~January 15, 2007~~

AUXILIARY

MAIN REQUEST

computing
mobile device printingof allowing a mobile device ^{<->} not having a printer driver to print on a printer,

1. A method in a computing environment ~~for facilitating accessing the computing resources of devices by other devices~~, the method comprising:
 - ~~at a first computing device (202, 400, 500) associated with at least one identity,~~
the mobile device ^{<->} discovering ^{another} ~~a second~~ computing device (212, ~~402, 404, 502, 504~~) ~~associated with the at least one identity,~~
equipped with a printer driver (406) for said printer;
 - ~~at the first device,~~ the mobile device ^{<->} obtaining data (209B) related to the ~~second~~ ^{another computing} device, the data including ~~resource~~ ^{printer driver} information of the ~~second~~ ^{another computing} device formatted according to a schema understood by the ~~first~~ ^{mobile <->} device;
 - the ~~first~~ ^{mobile <->} device interpreting the data to access ~~a resource (R_{B1}-R_{Bn})~~ ^{the printer driver} of the ~~second~~ ^{another computing} device; and
 - the ~~first~~ ^{mobile <->} device performing a ~~computing~~ ^{printing} task including ~~accessing the resource of the another second device to perform at least part of the computing task.~~ ^{providing printing data via the printer driver.}
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the ~~first~~ ^{mobile <->} device to the ~~another second~~ ^{another computing} device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the ~~first~~ ^{mobile <->} device to the ~~second~~ ^{another computing} device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

<computing>



• PATENTEN
85328 München
• +49 89 2399-0
• +49 89 2399-4444
FAX: +49 89 2399-4444

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 2

Dirección General 2

Direction Générale 2

Telephone numbers: Search at The Hague

Primary Examiner Substantive examination

Formalities Officer Assistant Formalities and other matters

Grünecker, Kinkeldey,
Stockmair & Schwannhäuser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



Application No.
03 009 717.4 - 1244

Ref.
EP27314-034/df

Date
29.01.2007

Applicant
MICROSOFT CORPORATION

Result of consultation

A copy of the result of consultation of 23.01.2007 is enclosed for your information.



Goya, Jesus
For the Examining Division

Enclosure(s): Copy of result of consultation (Form 2036)
Copy of the patent application US5748890

Europäisches Patentamt
GD2

European Patent Office
DG2

Office européen des brevets
DG2

Application No. :

03 009 717.4

Consultation by telephone with the applicant / representative

Despatch for information

Participants

Applicant: Microsoft Corporation
Representative: Mr Eickelkamp
Member(s) of the
Examining Division: Goya, Jesus

Result of consultation

During the telephone conversation with Mr Eickelkamp the examiner expressed the view of the Examining Division in respect to the amendments filed on 15.01.2007 by the applicant for a main and 5 auxiliary requests. The following points were discussed:

- .. The amendments according to the Examining Division are all related to the idea of access control for a first computer that wants to access the resources of a second computer. This according to the amendments filed by the applicant is done by associating an identity to a first computer and matching this identity to the second computer. The main and auxiliary requests develop this idea, in some of the auxiliary requests the identity is associated to a resource, in other auxiliary request the identity determines the extend of access to the resource, in other the schema is resource-based.
- .. It is the opinion of the Examining Division that the idea of resource access control mechanisms is a well-known feature in the field of network design and that the skilled person, confronted with the problem of how to secure the access to resources would find in the state of the art documents pointing at this problem and its solution. As an example document US5748890 (abstract; column 2, line 33 to column 3, line 60; column 6, lines 5-14 and lines 42-50; column 7, lines 5-21; column 9, line 58 to column 10, line 2) discloses such problem and the same solutions (in the various forms as for the different main and auxiliary requests).

The examiner communicated to the applicant that the Examining Division was of the opinion that the mentioned requests (main and 5 auxiliary requests) are not allowable due to lack of inventive step (Article 56 EPC) of claims 1, 6, 7 and 17. Oral proceedings will take place on the 15th February.



Date 29.01.2007

Sheet 2

Application No.: 03 009 717.4

23.01.2007

Date

Goya, Jesus

Examiner

Enclosure(s):
Copy of the patent application US5748890

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER

ANWALTSSOZIELTÄT

EPO - Munich
40

15. Jan. 2007

GKS & S MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

European Patent Office
Erhardtstraße 27

80298 München

o.p. 15.02.07!

RECHTSANWÄLTE LAWYERS

MÜNCHEN
DR. HELMUT FICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRODER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANK, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA BERTRAM, LL.M.
KARIN LOCHNER
BABETT ERTLE
CHRISTINE NEUHIERL
SABINE PRÜCKNER
CORNELIA SCHMITT
BERNHARD MEHNERT, DIP.LING.
PETRA LÜBBE
DR. SONJA BRÖTJE
SONJA SCHÄFFLER
DR. HOLGER GAUSS
DR. NICOLAS SCHMITZ

PATENTANWÄLTE EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOB
HANS HILGERS
ANNELE EHNOLD
THOMAS SCHUSTER
DR. KLARA GOLDSACH
MARTIN AUERHANGEL
GOTTFRIED KUTZSCH
DR. HEIKE VOGELSANG-WENKE
REINHARD KNAUER
DIETMAR KUHLE
DR. FRANZ-JOSEF ZIMMER
BETTINA K. REICHELT
DR. ANTON K. PFÄU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (L'ÉPAP) M.S.
BERND ROTHHEIMEL
THOMAS W. LAUSENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS ECKELKAMP
DR. CARLO TORTI
DR. PETER MILTENY
DR. MORITZ-OFFER
EUROPEAN PATENT ATTORNEYS

PATENTANWÄLTE EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED BÖNING
DR. PATRICK ERKENS
KÖLN
DR. MARTIN DROPMANN
CHEMNITZ
DR. CARVEN STEINIGER
PATENTANWALT
ALICANTE
VOLFARD MEISTER
DR. ANDREAS TONKA
RECHTSANWALT
OF COUNSEL
DR. HENNING MEYER-PLATH
—
DR. WILFRIED STOCKMAYER
(1996)

IMP ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.

DATUM / DATE

EP27314TE109kle

January 15, 2007

European Patent Application No. 03 009 717.4 - 1244
Applicant: Microsoft Corporation

In response to the Summons to attend oral proceedings dated October 18, 2006:

Submitted herewith are:

A main request comprising claims 1 to 29, and

Five auxiliary requests, i.e. first to fifth auxiliary requests, each comprising claims 1 to 29.

Referring to the claims of the main request, new claim 1 is based on previous claims 1 and 7. New claims 2 to 5 correspond to previous claims 2 to 5, respectively. New claim 6 is based on previous claims 6 and 7. New claims 7 to 29 correspond to previous claims 7 to 29, where a typographical error has been corrected in claim 22.

Referring to the claim set of the first auxiliary request, new claims 1 to 29 correspond to the claims of the main request, where new claims 1, 6 and 7 have been supplemented with the feature of "the at least one identity identifies a user of the first computing device and the

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
http://www.grunecker.de
e-mail: info@grunecker.de

DEUTSCHE BANK MÜNCHEN
No. 17 51734
BLZ 700 700 10
SWIFT: DEUTDE33

second computing device", supported by the description on page 4, lines 11 to 14 and page 19, lines 14 to 19.

Referring to the claim set of the second auxiliary request, new claims 1 to 29 correspond to the claims of the first auxiliary request, where new claims 1, 6 and 7 have been supplemented with a feature. In detail, the feature of the resource being associated with the identity has been supplemented and is supported by the description on page 34, line 22 to page 35, line 5. The feature is also supported by previous claim 6.

Referring to the third auxiliary request, new claims 1 to 29 correspond to the claims of the second auxiliary request, where new claims 1, 6 and 7 have been further amended. This amendment is a feature supported by the description on page 20, lines 12 to 14. In particular, the feature defines that the user's identification determines the extent to which the resource may be accessed.

Referring to the fourth auxiliary request, new claims 1 to 29 correspond to the claims of the second auxiliary request, where new claims 1, 6 and 7 additionally define that only the computing task performed by the first computing device is authorized to access the resource of the second device. This feature finds support in the description on page 19, line 20 to page 20, line 1.

Referring to the fifth auxiliary request, new claims 1 to 29 correspond to the claims of the second auxiliary request, where new claims 1, 6 and 7 further define the schema as being resource-based, listing multiple devices for each resource. This supplemented feature finds support in the description on page 24, lines 3 to 5.

With respect to the main request and, in particular, the subject matter of independent claims 1, 6 and 7, the Examining Division is kindly requested to consider the following.

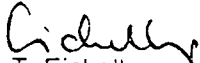
The new claims now define that the first and second computing devices are each associated with at least one identity. The subject matter of the independent claims differs from the disclosure of either document D1 or D2 at least in that each of the first and second computing devices is associated with the at least one identity. With respect to section 2.1 of the Summons and the last letter of reply dated March 22, 2006, it is respectfully submitted

that the feature defined by the independent claims of interpreting data to access a resource of the second computing device to perform a computing task including accessing the resource is believed to be novel over the disclosure of documents D1 and D2. In detail, the Examining Division has stated that the word "access" or "accessing" is interpreted as sending a request to the second computing device to *start* using its resources. In particular, sending a request to the second computing device to start using its resources appears to differ from accessing resources to perform a computing task. To perform a computing task, the resources are involved in the performing of the computing task, while sending a request can be seen as different computing tasks. In detail, the first task would be the sending from the first computing device, while the second task would be the receiving by the second computing device. Thus, the feature of accessing a resource to perform a computing task as defined by the present invention appears to refer to one computing task, which is performed on one computer, accessing the resource of another computer. This differs from sending and receiving a request from a first to a second computing device, since the task of sending the request would be completed while the received request is computed by the second device. Although the Examining Division states that the request is to start using the second device's resources, this appears not to be a single computing task as defined by the present invention, but rather seems to be the purpose of the request. The subject matter of the independent claims is therefore considered to be novel over documents D1 and D2 for at least the above discussed features.

In addition, since the present invention appears to be novel over documents D1 and D2, the Examining Division is kindly requested to reconsider the argumentations outlined in the last letter of reply dated March 22, 2006, beginning on page 2, third paragraph to page 4, third paragraph. In particular, neither D1 nor D2 render obvious the accessing of a resource of one computer by a different computer which is performing a task including accessing this resource. In addition, neither of the prior art references suggest a first computing device and a second computing device which are both associated with one identity. Thus, the first and second computing devices are linked to each other by being associated with the identity.

While document D1 relates to a load sharing system having a plurality of computers which are operating in parallel, D2 relates to a computer system also including a plurality of computers, i.e. terminals. Neither of these documents is related to a system comprising at

least two computing devices where each computing device is associated with the same identity. Thus, the skilled person, confronted with the object of providing a method and system to discover and expose computing resources of various computing devices to one another for cross-resource usage, seems to have a variety of measures at hand. However, the combination of a system having a first device associated with an identity and a second device associated with the identity, and the fact that the first device is performing a computing task including accessing a resource of the second device to perform at least part of the computing task, is not rendered obvious to the skilled person by either of documents D1, D2 or a combination thereof. The subject matter of the new independent claims therefore involves an inventive step.


T. Eickelkamp

Encls.:

Six new claim sets (main request and first to fifth auxiliary requests)

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

MAIN REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;
 - at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;
 - the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device; and
 - the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task.
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device from the information, the second device being associated with the at least one identity;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
 - a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task.
8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.

12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:
 - a first field for identifying a type of resource;
 - a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;
 - a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.
18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.

20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

FIRST AUXILIARY REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;
 - at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;
 - the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device; and
 - the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task;wherein the at least one identity identifies a user of the first computing device and the second computing device.
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
- at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device from the information, the second device being associated with the at least one identity;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
- wherein the identity identifies a user of the first computing device and the second device.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
- a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task;
- wherein the identity identifies a user of the first computing device and the second device.
8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.

10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.
12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:
 - a first field for identifying a type of resource;
 - a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;
 - a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

SECOND AUXILIARY REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;
 - at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;
 - the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device, the resource being associated with the identity; and
 - the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task;wherein the identity identifies a user of the first computing device and the second computing device.
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).

6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
- at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device associated with the at least one identity from the information;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
- wherein the identity identifies a user of the first computing device and the second device, and wherein the resource of the at least one other device is associated with the identity.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
- a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema, the resource being associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task;
- wherein the identity identifies a user of the first computing device and the second device.
8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).

9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.
12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:
 - a first field for identifying a type of resource;
 - a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).

29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

THIRD AUXILIARY REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:

at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;

at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;

the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device, the resource being associated with the identity; and

the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task;

wherein the identity identifies a user of the first computing device and the second computing device, and wherein the user's identification determines the extent to which the resource may be accessed.
2. The method of claim 1 further comprising:

communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:

communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).

5. The method of claim 4 wherein the credentials are provided by an authentication service (220).
6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device associated with the at least one identity from the information;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
 - wherein the resource of the at least one other device is associated with the identity, wherein the identity identifies a user of the first computing device and the second device, and wherein the user's identification determines the extent to which the resource may be accessed.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
 - a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema, the resource being associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task;

wherein the identity identifies a user of the first computing device and the second device, and wherein the user's identification determines the extent to which the resource may be accessed.

8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.
12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:

- a first field for identifying a type of resource;
- a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;
- a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.
18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
 19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
 20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
 21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
 22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
 23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
 24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
 25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
 26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
 27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).

28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

FOURTH AUXILIARY REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;
 - at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;
 - the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device, the resource being associated with the identity; and
 - the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
 - wherein the identity identifies a user of the first computing device and the second computing device, and wherein only the computing task performed by the first computing device is authorized to access the resource of the second device.
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).

5. The method of claim 4 wherein the credentials are provided by an authentication service (220).
6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device associated with the at least one identity from the information;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
 - wherein the resource of the at least one other device is associated with the identity, wherein the identity identifies a user of the first computing device and the second device, and wherein only the computing task performed by the first computing device is authorized to access the resource of the second device.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
 - a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema, the resource being associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task;

wherein the identity identifies a user of the first computing device and the second device, and wherein only the computing task performed by the first computing device is authorized to access the resource of the second device.

8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.
12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:

a first field for identifying a type of resource;

a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).

28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Application No.: 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION
Our Ref.: EP27314TE109
Date: January 15, 2007

FIFTH AUXILIARY REQUEST

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity, discovering a second computing device (212, 402, 404, 502, 504) associated with the at least one identity;
 - at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;
 - the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device, the resource being associated with the identity; and
 - the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task;wherein the identity identifies a user of the first computing device and the second computing device, and wherein the schema is resource-based listing multiple devices for each resource.
2. The method of claim 1 further comprising:
 - communicating (600) a request to use the resource from the first device to the second device.
3. The method of claim 1 or 2 further comprising:
 - communicating identity information from the first device to the second device.
4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).

5. The method of claim 4 wherein the credentials are provided by an authentication service (220).
6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:
 - at a first computing device (202, 400, 500) associated with at least one identity:
 - locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;
 - selecting a resource of a second device associated with the at least one identity from the information;
 - accessing the resource of the second device; and
 - performing a computing task including accessing the resource of the second device to perform at least part of the computing task;
 - wherein the identity identifies a user of the first computing device and the second device, wherein the resource of the at least one other device is associated with the identity, and wherein the schema is resource-based listing multiple devices for each resource.
7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:
 - a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema, the resource being associated with the at least one identity;
 - a second device (202, 400, 500), the second device configured to request access to the resource; and
 - the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task;

wherein the identity identifies a user of the first computing device and the second device, and wherein the schema is resource-based listing multiple devices for each resource.

8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
11. The system of claim 7 wherein the resource comprises an input method.
12. The system of claim 7 wherein the resource comprises an output method.
13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:

a first field for identifying a type of resource;

a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.
25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).

28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

Administration des postes des Pays-Bas
voor kantoor van afzending in te vullen

Kantoor van terpostbezorging
Bureau de dépôt

Datum
Date 18 10 2005

Bestemmingsadres van de zending
Destinataire de l'envoi
Gründer, Kindeldays
Stechmair und Schwannhäuser
Maximilianstrasse 58
80538 München
ALLEMAGNE

Aard van de zending/Nature de l'envoi

☒ Briefpost
Lettre

☐ Pakket Nr.
Colis No.

☒ Aangetekend
Recommandé

Waarde-aangifte
Valeur déclarée 23.10.00

☐ Munteenheid/Unité monétaire
Bedrag/Montant

Aangetekend

Recommandé
NL

3S ROEB 1631545



RV 88 582 516 1DE

R

Bericht van ontvangst/uitbetaling/bijbeschrijving CN 07
AVIS de réception/de paiement/d'inscription

A.R.

Service des postes
Stempel van het
kantoor van terug-
zending van het be-
timbre du bureau
renvoyant l'avis

EPA EPO OEB
tel.: (070) 3404616

31 OCT 2006

Mailroom IV

Terugzenden aan (in te vullen door de afz.)
Renvoyer à (à remplir par l'expéditeur)

Naam of firma
Nom ou raison sociale

European Patent Office

Straat en huisnummer
Rue et n°

PB 5818 - Patentlaan 2

Postcode en plaats
Localité

2280 HV - Rijswijk

Land
Pays

Netherlands

Post BV Handelsregister: (Versie 0301) - 020432
Haglanden nr. 27124700

616

092

Bitte umgehend zurücksenden!
Please return immediately!
Prière de renvoyer immédiatement !

EPO - Munich
40

24. Okt. 2006

Empfangsbescheinigung über den Zugang des nachstehend bezeichneten Schriftstücks
Acknowledgement of receipt of the document specified below
Récépissé du document spécifié ci-après

Anmeldenummer / Beschwerdeaktenzeichen Application number / appeal case number N° de la demande / n° de référence du recours	Datum Date	EPA/EPO/OEB Form
03009717.4-1244	18. 10. 2006	Oral Proc. 2008

For the President, RINGANG & SCHWANHAUSSER
ANWALTSSOZIOZETAT
23. Okt. 2006
EPA/OEB Form 2535 G5.05
TEAM
RINGANG - RECEIVED

Unterschrift / Signature

Empfangsberechtigter / authorised recipient /
le destinataire ou la personne dûment mandatée

Zur gefälligen Beachtung

Unter Bezugnahme auf die Mitteilung
im ABl. des EPA 11/1991, 577, wird
gebeten, die Empfangsbescheinigung
mit **Empfangsdatum** und **Unterschrift**
zu versehen und **umgehend** an das EPA
zurückzusenden.

Please note

With reference to the Notice in EPO OJ
11/1991, 577, you are requested to **date**
and **sign** the acknowledgement of receipt
and return it to the EPO **immediately**.

Nota bene

Conformément à la communication parue
au JO OEB 11/1991, 577, vous êtes prié
de porter sur le récépissé la **date d'arrivée**
du document, ainsi que votre **signature**
et de le renvoyer **immédiatement** à l'OEB.

An das

Europäische Patentamt

80298 München

ALLEMAGNE / GERMANY /
DEUTSCHLAND



Bescheid/Protokoll (Anlage)

Datum
Date 18.10.2006
Date

Communication/Minutes (Annex)

Blatt
Sheet 1
Feuille

Notification/Procès-verbal (Annexe)

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

The examination is being carried out on the **following application documents**:

Description, Pages

1-3, 5-19, 21-36, 38, as originally filed
39

4, 4b, 20, 37, 40 received on 10.01.2005 with letter of 05.01.2005

4a received on 22.03.2006 with letter of 22.03.2006

Claims, Numbers

1-29 received on 22.03.2006 with letter of 22.03.2006

Drawings, Sheets

1/7-7/7 as originally filed

- 1 The applicant is summoned to attend the oral proceedings on the 15.02.2007 at 10.00h. The following observations are intended to assist the applicant in preparing the forthcoming Oral Proceedings and shall not be binding for the Examining Division (see Guidelines, E-III-6).

The issues to be discussed during the oral proceedings mainly being the following:

Article 54(1) and (2) EPC with regard to the independent **claims 1, 6 and 7** in respect of document D1, and also in respect of document D2; and Article 56 EPC with regard to the independent **claim 17** in respect of document D1.



2 The applicant, with his letter of 22 March 2006, has not provided arguments that could convince the Examining Division on the patentability of the subject-matter of the application. The present independent **claims 1, 6, 7 and 17** remain unamended as to their substance and therefore, the objections of the previous communication by the examiner on 07.09.2005 are maintained. In addition, the following is noted by the examiner:

2.1 The mentioned claims have been amended with the feature "... for facilitating accessing the computing resources of devices by other devices ...", also the expression "computing task including using the resource" has been replaced by "computing task including accessing the resource". It is clear from document D1 (=US-A-5898870) (abstract), as well as from D2 (=US-A-6078955) or D3 (=US5640569), that the client computers select a remote computer for executing a work and consequently access the resources of a remote computer for such execution of a work.

The word "access" or "accessing" is interpreted by the Examiner as sending a request to the second computing device to start using its resources. This is also disclosed in D1 (column 3, lines 46-50).

2.2 The applicant mentions in his letter that the application discloses the interpretation of resource information data by the first computing device to access a resource of another computer. According to the applicant, D1 only teaches the selection of the optimum system for a particular job.

The Examining Division disagrees with the view of the applicant, as from D1 (column 3, lines 25-50) the first computing device obtains information about resources of other computers ("resource utilization-state management table") and compares this information with data related to the job that needs to be executed. It is, therefore, implicit from D1 that the resource information is interpreted by the first computing device prior to accessing such resource.

2.3 According to the letter of the applicant, the feature of "the first device performing a computing task including accessing the resource of the second device to perform at

Datum
Date 18.10.2006Blatt
Sheet 3
FeuilleAnmelde-Nr.:
Application No. 03 009 717.4
Demande n°:

least part of the computing task" is not disclosed in the prior art documents. The applicant considers that in D1 the first computing device does not perform any computing task, the whole computing task is performed in the second computing device.

The Examining Division agrees with the applicant's interpretation of D1: D1 does not explicitly disclose the execution of part of the computing task in the first computing device, but the whole computing task is executed in the second computing device.

However, the Examining Division considers that the current wording of the claim "at least part of the computing task" also comprises the particular case in which the whole computing task is executed in the second computing device. This particular case corresponds to one of the embodiments in the current application, description, page 30, second paragraph, where a mobile device accesses printing resources, shows the case of execution of the whole computing task in the second computing device. Document D1 discloses this particular case of execution of the whole computing task in the second computing device, and therefore anticipates the feature in question.

The applicant is informed that even if the claim were amended in a way that part of the computing task is performed in the first computing device and another part is performed in the second computing device, the subject-matter of such a claim would not be inventive over the disclosure of D1. For example one of the commonly known cases is that the execution of a computing task in a remote second computing device involves returning a result of such execution to the first computing device, and that such result would be further processed by the first computing device, for example in order to display the result (see for example D2, claims 4-6. The execution of a computing task in a second computing device implies the modification of a file in the first computing device, which is considered part of the computing task that is executed in the first computing device).

- 2.4 According to the letter of the applicant, the invention of the application differs from the disclosure of D1 in that the first computing device receives and interprets resource information data from a second computing device and afterwards accesses



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 18.10.2006

Blatt
Sheet 4
Feuille

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

a resource of the second computing device.

The Examining Division agrees that in D1 the information about the second computing device is not obtained from the second computing device but from a central computing device.

However, claim 1 of the application does only specify that the first computing device obtains the resource information data, without specifying whether it is received from the second computing device. Moreover, one of the embodiments of the current application, page 22, last paragraph, discloses that the resource information data is centralized in a device that can be accessed by the first computer device. This corresponds to the disclosure of D1.

The applicant is informed that even if the independent claims were amended to present the embodiment in which the first computing device receives the resource information from the second computing device, it would be considered an implementation-related design option that would be obvious for the skilled person in the field of data networking.

- 2.5 The applicant in his letter mentions that the documents D1, D2 or D3 would not be available to the skilled person in order to solve an objective problem indicated by the applicant. It is the opinion of the Examining Division that this is not a relevant argument for the assessment of claims 1, 6 and 7, as these independent claims are considered as not novel over the disclosure of such documents.

Furthermore, for the assessment of claim 17, the Examining Division is of the opinion that the above mentioned documents belong to the same technical domain of the application, namely, the use of computers resources in a distributed architecture, and as from the previous communication from the examiner, disclose the same technical problem and solve it in the same way.

- 2.6 It is furthermore noted that, in addition to the objection of the previous communication by the examiner on 07.09.2005, point 2.1, regarding the lack of novelty of claim 1 over the disclosure of D1, the wording of the independent claim 1 is at present such



Datum
Date 18.10.2006
Date

Blatt
Sheet 5
Feuille

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

that its scope even covers well known scenarios for the skilled person in data networking, such as the remote access and execution of files. In this scenario a user is able to retrieve information about resources (files, programs) stored in other computers, and according to this information (which can be the identifier of a file or program, size in memory, availability, etc) the user can decide to access these resource (execution of a file, visualization of an image, playing of a song, etc). This embodiment is also disclosed in the description of the application, page 36.

The Examining Division is of the opinion that such embodiment comprised within the scope of the independent claims is commonly known in the art, and in particular is known from D2 (see abstract); consequently, claim 1 of the application is not allowable, Article 52(1) EPC, due to lack of novelty, Article 54(1) and (2) EPC, also for this reason.

- 3 The dependent claims do not introduce any feature, which in combination with the subject-matter of the independent claims would render the application allowable under Article 52(1) EPC (the applicant is referred to the first communication by the examiner on 30.08.2004).
- 4 The attention of the applicant is also drawn to the fact that the last date for written submissions and/or amendments (which must comply with Article 123(2) EPC and Rule 36 EPC) is 26.5.2006 and also to the provisions of Rule 71(2) EPC and Rule 71(a) EPC.
- 5 If the applicant should decide to renounce to the Oral Proceedings, the applicant is kindly asked to inform the EPO at the earliest possible.

J. GOYA

623

Bitte umgehend zurücksenden!
Please return immediately!
Prière de renvoyer immédiatement !

Empfangsbescheinigung über den Zugang des nachstehend bezeichneten Schriftstücks
Acknowledgement of receipt of the document specified below
Récépissé du document spécifié ci-après

Anmeldenummer / Beschwerdeaktenzeichen Application number / appeal case number N° de la demande / n° de référence du recours	Datum Date	EPA/EPO/OEB Form
03009717-4-1244	18. 10. 2006	Oral Proc. 2008

Empfangen am / Received on / Reçu le

Unterschrift/Signature

EPA/EPO/OEB Form 2936 05.05

Empfangsberechtigt / authorised recipient /
le destinataire ou la personne dûment mandatée

Administration des postes / Postamt
Bureau de poste / Ufficio postale

Bericht van ontvangst/uitbetaling/bijdrifving - CN 07
AVIS de réception/de paiement/d'inscription

A.R.

Service des postes
Stempel van het
kantoor van de
zending van het bericht
Inmiddels 60 miljoen
700.000 miljoen

Naam of firma
Nom ou raison sociale

Straat en huisnummer
Rue et n°

Postcode en plaats
Localité

Land
Pays

European Patent Office

PB 5818 - Patentlaan 2

2280 HV - Rijswijk

Netherlands

Op plaats van bestemming in te vullen
A remplir à destination

De hierbovenvermelde zending is naar behoren
L'envoi mentionné ci-dessus a été dûment

Ontvangen
remis

Uitbetaald
payé

Bijgeschreven op giro
Inscrit en CCP

Datum en handtekening/Date et signature

Afhankelijk van de voorschriften in het land van bestemming wordt het bericht getekend
door de geadresseerde of zijn gemachtigde of door de postmedewerker

03009717-4-1244

297



P.B. 5518 - Patentamt
2250 HV Rijswijk (ZH)
☎ (070) 3 40 20 40
FAX (070) 3 40 30 18

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 1

Directorate General 1

Direction générale 1

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date

18-10-2006

Reference

EP27314-034/df

Application No./Patent No.

03009717.4 - 1244

Applicant/Proprietor

MICROSOFT CORPORATION

Summons to attend oral proceedings pursuant to Rule 71(1) EPC

You are hereby summoned to attend oral proceedings arranged in connection with the above-mentioned European patent application.

The matters to be discussed are set out in the communication accompanying this summons (EPO Form 2906).

The oral proceedings, which will not be public, will take place before the examining division

on 15.02.07 at 10.00 hrs in Room 0D.06, EPO
Rijswijk, ~~Patentamt~~ NL-2288 EE Rijswijk (ZH)
Visseringlaan 19-23

No changes to the date of the oral proceedings can be made, except on serious grounds (see OJ EPO 10/2000, 456).

If you do not appear as summoned, the oral proceedings may continue without you (R. 71(2) EPC). Your attention is drawn to Rule 2 EPC, regarding the language of the oral proceedings, and to the OJ EPO 9/1991, 489, concerning the filing of authorisations for company employees and lawyers acting as representatives before the EPO.

The final date for making written submissions and/or amendments (Rule 71a EPC), is 15.01.07.

You are requested to report in good time beforehand to the porter in the EPO foyer. Room _____ is available as waiting room.

Parking is available free of charge in the underground car park (see map enclosed).

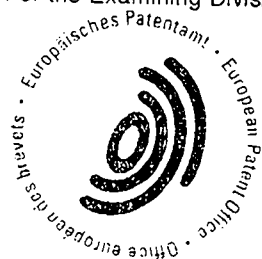
1st Examiner:
Goya Alonso J

2nd Member:
Peeters D

Chairman:
Kesting V

For the Examining Division

Annexes:
Confirmation of receipt (Form 2936)
Communication (EPO Form 2906)



Registered letter with advice of delivery
EPC Form 2008 05.05 [ORAL03=AD06] 12.10.06

to EPO postal service: 12.10.06

624

298

625

Application No.: 03 009 717.4 ①

Preparation for oral proceedings - Instructions to Support Service

Oral proceedings are to be held in connection with the above patent application

1. The matters to be discussed are set out in the annex (Form 2906)
2. Dispatch the summons using Form 2008/2310 for the parties to attend on:

Day 15.02.2007 Time 10h

BDSY DHSEVI

Room booked	
Room D06	

ORAL 1,2,3 coded

A. Groeneveld
Date 12 10 2006 Initials

- 2.1 Parties' submissions in preparation for the oral proceedings, if any, should be made no later than

1 month(s)

before the date of the oral proceedings
(transfer to Form 2008.1 / 2310.1)

- 2.2 Encode ORAL(4) ②

A. Groeneveld
coded 12 10 2006
Date Initials

- 2.3 Dispatch Form 2008.7 / 2310.7 to division

A. Groeneveld
Date 12 10 2006 Initials

3. ☒ Arrange for the following special equipment to be provided in the conference room:

Coffee, tea, water

A. Groeneveld
Date 12 10 2006 Initials

4. Request language service to provide simultaneous interpretation facilities as necessary

Date Initials

299

626

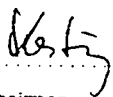
5. Return the dossier to primary examiner
after dispatching the summons

.....
Date Initials

6. Check that summons has been received
(receipt - white / advice of delivery - red)

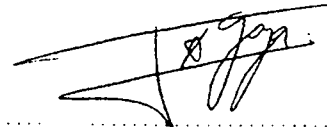
7. 15 days before the oral proceedings:
- dispatch the dossier to the primary examiner.

9. 10. 2006
Date


Chairman


2nd examiner

Dinh
Peeters


Goya, Jesus
1st examiner

.....
Legal member

300

627

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSE EPO - Munich
ANWALTSSOZIELTÄT

2.2. März 2006

GKS & S. MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

European Patent Office
Erhardtstraße 27

80298 München

RECHTSANWÄLTE
LAWYERS

MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRÖDER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANKÉ, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELMIRA BERTRAM, LL.M.
KARIN LOCHNER
BABBETT ERTLE
CHRISTINE NEUMIERL
SABINE PRÜCKNER
CORNELIA SCHMITT
BERNHARD MEHNERT, DIPL. ING.
PETRA LÜBBE
DR. SONJA BRÖTJE

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOB
HANS WILGERS
DR. HENNING MEYER-PLATH
ANNELE EHNOLD
THOMAS SCHUSTER
DR. KARA GOLDBACH
MARTIN AUFENANGER
GOTTFRIED KLITZSCH
DR. HEIKE VOGELANG-WENKE
REINHARD KNAUER
DIETMAR KLHHL
DR. FRANZ-JOSEF ZIMMER
BEITINA K. REICHELT
DR. ANTON K. PFÄU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (U of PA) M.S.
BERND ROTHMEL
THOMAS W. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HANMER
DR. THOMAS EICKELKAMP
DR. CARLO TORTI
EUROPEAN PATENT ATTORNEY

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED BÖNING
DR. PATRICK ERB, M.S. (U of PA)
KÖLN
DR. MARTIN DOPPMANN
CHEMNITZ
DR. CARMEN STEINIGER
PATENTANWALT
ALICANTE
WOLFGANG MEISTER
DR. ANDREAS TORKA
RECHTSANWALT
DR. WILFRIED STOCKMAIR
(1936)

IHR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.
EP27314TE109rey

DATUM / DATE
March 22, 2006

European Patent Application No. 03009717.4-1244
Title: IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE
RESOURCES
Applicant: Microsoft Corporation

In response to the Official Communication dated September 12, 2005:

Submitted herewith are:

New claims 1 to 29 to replace all claims presently on file; and

New description page 4a to replace page 4a presently on file.

New claim 1 is based on previous claim 1 and is further supported by the description on page 37, lines 5 to 7. New claims 2 to 5 correspond to previous claims 2 to 5, respectively. New claims 6 and 7 are based on previous claims 6 and 7, respectively, and are further supported by the description on page 37, lines 5 to 7. New claims 8 to 16 correspond to previous claims 8 to 16, respectively. New claim 17 is based on previous claim 17. New claims 18 to 29 correspond to previous claims 18 to 29, respectively.

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSE
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
http://www.grunecker.de
e-mail: info@grunecker.de

DEUTSCHE BANK MÜNCHEN
No. 17 51734
BLZ 750 700 10
SWIFT: DEUT 33 33

3ed

The number of independent claims has been reduced by converting previous independent claim 17 to a dependent claim. The last feature of the remaining independent claims 1, 6 and 7 has been amended by replacing the expression "including using the resource" by the phrase "including accessing the resource". This amendment finds support, for example, in claim 6, defining the step of accessing the resource, and is further supported by the description at various places. For instance, on page 36, lines 17 to 24, an implementation is described which includes an application program and various distributed resources and in addition, on page 37, lines 5 to 7, it is described that the resources on different computer systems are accessed to essentially form a single functioning system. The preamble of the independent claims 1, 6 and 7 has been supplemented with a feature emphasizing the accessing of the resources.

The description has been amended by revising the object of the invention.

Discussing now the prior art objections raised in section 2 of the Official Communication, the Examining Division is kindly requested to consider the following, in particular in view of the new independent claims.

In detail, document D1 relates to a parallel computer system including a computer group containing a plurality of computers, and an operation management mechanism. D1 further describes a method for controlling load sharing in the parallel computer system. On the other hand, the subject matter of the new independent claims 1, 6 and 7 is directed to a system and method for facilitating accessing resources of devices by other devices. In other words, while document D1 describes a system where a requested job is executed by a selected optimum system (see in particular D1, column 9, lines 13 to 17), the present invention appears to describe a system where a device performs a computing task by including the resource of a different device.

Thus, the subject matter of new independent claims 1, 6 and 7 differs from the disclosure of document D1 at least in that a first computing device interprets data received from a second computing device to **access** a resource of the second computing device, and further in that the first device performs a computing task including accessing this resource.

When starting from the background of the invention, the object of the invention is to provide a method and system to discover and expose computing resources of various computing devices to one another for cross-resource usage (see present application, page 4, lines 1 to 3).

The skilled person, confronted with the above objective technical problem, would primarily look for a solution to the problem in the field of common computing devices, or maybe in the field of distributed computing.

Since document D1 relates to a load sharing system which includes a plurality of somewhat similar computers which are operating in parallel, and which are connected to a high-speed connection mechanism serving as an operation manager between the parallel computers, it does not appear naturally to the skilled person to adapt a load sharing system including a plurality of parallel computing systems for a possible solution to the above mentioned problem. However, even if assuming that the skilled person would consider the teaching of document D1 as relevant, the skilled person finds the following load sharing control method. Firstly, resource utilization target values by work for the plurality of parallel computers are set. Further, resource utilization states by work are collected for the plurality of parallel computers, and the operation management mechanism is thereby informed of the resource utilization states. Finally, a computer is selected to execute a newly requested work process from the plurality of parallel computers on the basis of the differences between resource utilization target parameter values by work and current values of a parameter indicating the reported resource utilization states by work. The newly requested work process is executed in the selected computer (see document D1, column 3, lines 1 to 14). Thus, the skilled person taking into account the teaching of document D1 would employ a centralized system, such as a server, to store information regarding the resource utilization of the plurality of computers.

Further, based on the collected information stored on the central server, one of the computing devices would be selected to execute a requested work process. In particular, D1 teaches in column 9, lines 13 to 17 to select the optimum system for executing a requested job which can be either the own computer system or another computer system.

Although D1 seems to suggest a finer granularity of the type of job to be executed, each job or type of job is performed by the optimum system (see column 6, lines 59 to 65).

In other words, D1 teaches to select one computer out of a plurality of parallel computers to execute a job, whereby the selected computer provides the optimum resources for that job. In addition, a resource utilization state management table is employed to store utilization ratios for certain resources for the determination of the optimum system.

However, this is not what is taught by the present invention. In particular, the present invention defines a method and system where one computing device can obtain data related to another computing device, and further may interpret the data to access a resource. While D1 only suggests to select the optimum system for a particular job, in the present application a computing device can interpret data exchanged with another computing device to access a resource of the other computing device. Moreover, the first computing device performs a computing task including accessing the resource of the second device to perform at least part of the task. Thus, D1 does not suggest the accessing of a resource of one computer by a different computer which is performing a task including and accessing this resource. Therefore, the subject matter of independent claims 1, 6 and 7 involves an inventive step.

As the application therefore now appears to be in good condition for grant of a patent, the Examining Division is respectfully requested to issue a Communication according to Rule 51(4) EPC as the next step of the proceedings.


(Koch)

Encl.
New claims 1 to 29
New description page 4a

Application No.: 03 009 717.4 - 1244

Applicant: MICROSOFT CORPORATION

Our Ref.: EP27314TE109kle

Date: March 22, 2006

CLAIMS

1. A method in a computing environment for facilitating accessing the computing resources of devices by other devices, the method comprising:

at a first computing device (202, 400, 500), discovering a second computing device (212, 402, 404, 502, 504);

at the first device, obtaining data (209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;

the first device interpreting the data to access a resource (R_{B1} - R_{Bn}) of the second device; and

the first device performing a computing task including accessing the resource of the second device to perform at least part of the computing task.

2. The method of claim 1 further comprising:

communicating (600) a request to use the resource from the first device to the second device.

3. The method of claim 1 or 2 further comprising:

communicating identity information from the first device to the second device.

4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).

5. The method of claim 4 wherein the credentials are provided by an authentication service (220).
6. A computer-readable medium having computer-executable instructions to perform a method for facilitating accessing the computing resources of devices by other devices, the method comprising:

at a first computing device (202, 400, 500):

locating information (209A, 209B) about a resource (R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;

selecting a resource of a second device from the information;

accessing the resource of the second device; and

performing a computing task including accessing the resource of the second device to perform at least part of the computing task.

7. A system in a computing environment for facilitating accessing the computing resources of devices by other devices, the system comprising:

a first computing device (212, 404, 504) associated with at least one identity, the first device including a resource (R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209B) about the resource according to a resource schema associated with the at least one identity;

a second device (202, 400, 500), the second device configured to request access to the resource; and

- the second device performing a computing task including accessing the resource on the first device to perform at least part of the computing task.
8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
 9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
 10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
 11. The system of claim 7 wherein the resource comprises an input method.
 12. The system of claim 7 wherein the resource comprises an output method.
 13. The system of one of claims 7 to 12 further comprising a third computing device (402, 502) connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
 14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
 15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
 16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
 17. A computer-readable medium for use in the method of one of claims 1 to 5, the computer readable medium having stored thereon a resource information

exchange data structure being suitable for providing resource information (209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:

a first field for identifying a type of resource;

a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.

25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

US-A-5 898 870 describes load sharing in a parallel computer system having a computer group constituted by a plurality of computers, and an operation management mechanism which is a computer for managing the operation of the computer group. Firstly, resource utilization target values by work are set in advance in the operation management mechanism, and subsequently the respective computers of the computer group inform the operation management mechanism of collected resource utilization states by work. In response to a request by a computer to newly start a work process, the resource utilization target value of the work process is compared with the resource utilization state thereof on a resource utilization state management table which is prepared in the initialization of the system at the operation management mechanism. Further, one of the computers is selected from the plurality of computers constituting the computer group on the basis of the result of the comparison as the optimum system and is prompted to execute the work process.

US-A-6 078 955 describes a method of controlling a computer system including a plurality of computers interconnected by a network and used by a plurality of users. The computer system includes at least a server, an administration server and a plurality of terminals. The administration server is responsible for managing all the computer resources and all the jobs performed on the computers as well as the entire user information in the computer system. The administration server collects and manages information in a system resource management table. Further, when a user logs in to the system at a terminal, the log-in information and the terminal information used for the log-in are sent to the administration server. After a processing request is sent from a terminal to the administration server, a server or terminal judged to be capable of handling a given process most efficiently is determined from the prevailing processing load of the servers in the network and the property and the location of the file and the program used for the intended process, and the place of execution of the process is notified to the particular terminal. The terminal then requests a computer corresponding with the notification to execute the process.

SUMMARY OF THE INVENTION

It is the object of the invention to provide a method and system to discover and expose computing resources of various computing devices to one another for cross-resource usage.

This object is solved by the invention as claimed in the independent claims.

Preferred embodiments are defined by the dependent claims.

637



P.B. 5818 • Postfach 2
2230 H.V. Rijswijk (ZH)
☎ (070) 3 40 20 40
FAX (070) 3 40 30 16

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 1

Directorate General 1

Direction générale 1

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



EPO Customer Services

Tel.: +31 (0)70 340 45 00

Date
15.02.06

Reference
EP27314-034/df

Application No./Patent No.
03009717.4 - 1244

Applicant/Proprietor
MICROSOFT CORPORATION

Extension of time limit pursuant to Rule 84 EPC

Examination procedure

With reference to your request, the time limit for replying to the communication dated 12.09.05 has been extended

by 2 months

to a total of 6 months

from the date of notification of the above-mentioned communication.

Please note: To the extent that your request exceeded the above extension, your request has been refused.

Note:

The granting of extensions to time limits is governed by the implementing Regulations to the EPC and the Guidelines for Examination in the EPO, part E-VIII, 1.6.

If no reply to the communication is received in due time, the European patent application will be deemed to be withdrawn (Article 96(3) EPC).

Examining Division



311

638

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER
ANWALTSSOZIENTÄT

EPO - Munich
23

23. Jan. 2006

GKS & S. MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

European Patent Office
Erhardtstraße 27

80298 München

RECHTSANWÄLTE
LAWYERS

MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRÖDER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANK, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA BERTRAM, LL.M.
KARIN LOCHNER
BABETT ERTLE
CHRISTINE NEUHIERL
SABINE PRUCKNER
CORNELIA SCHMITT
BERNHARD MEHNERT, DIPL. ING.
PETRA LÜBBE
DR. SONJA BRÖTJE

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOB
HANS HILGERS
DR. HENNING MEYER-PLATH
ANNELE EHNOLD
THOMAS SCHUSTER
DR. KILARA GOLDSACH
MARTIN AUFSNANGER
GOTTFRIED KLITZSCH
DR. HEIKE VOGEL-SANG-WEINKE
REINHARD KNAUER
DIETMAR KÜHL
DR. FRANZ-JOSEF ZIMMER
BETTINA K. REICHELT
DR. ANTON K. PFÄU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (J. of Pat.) M.S.
BERND ROTHMEL
THOMAS W. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS EICKELKAMP
DR. CARLO TORTI
EUROPEAN PATENT ATTORNEY

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED BONING
DR. PATRICK ERK, M.S. (J. of Pat.)
KÖLN
DR. MARTIN DROPMANN
CHEMNITZ
DR. CARMEN STEINIGER
PATENTANWÄLTIN
ALICANTE
WOLFGANG MEISTER
DR. ANDREAS TORKA
RECHTSANWALT
—
DR. WILFRIED STOCKMAIR
(1958)

HR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.
EP27314TE109kle

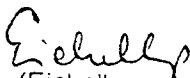
DATUM / DATE
January 23, 2006

European Patent Application No. 03 009 717.4 - 1244
Applicant: MICROSOFT CORPORATION

In response to the Official Communication dated September 12, 2005:

We herewith respectfully request to extend the term for replying the outstanding Official Communication by two months.

We have just received instructions from the applicant for a response to the pending Official Communication. However, after carefully studying said instructions, we discovered some portions which require clarification. Consequently, further correspondence with the applicant is necessary.


(Eickelkamp)

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
<http://www.grunecker.de>
e-mail: info@grunecker.de

DEUTSCHE BANK MÜNCHEN
No. 17 51734
BLZ 700 700 10
SWIFT: DEUT DE 33

639



EPA/EPO/OEB
D-80298 München
+49 89 2399-0
TX 523 656 epmu d
FAX +49 89 2399-4465

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 2

Directorate General 2

Direction Générale 2

Grünecker, Kinkeldey, Stockmair &
Schwanhäusser Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Telephone numbers: Branch at The Hague

Primary Examiner -31 70 340-2370
(substantive examination)

Formalities Officer / Assistant -31 70 340-3478
(Formalities and other matters)



Application No.
03 009 717 4 - 1244

Ref.
EP27314-034/df

Date
12.09.2005

Applicant
MICROSOFT CORPORATION

Communication pursuant to Article 96(2) EPC

The examination of the above-identified application has revealed that it does not meet the requirements of the European Patent Convention for the reasons enclosed herewith. If the deficiencies indicated are not rectified the application may be refused pursuant to Article 97(1) EPC.

You are invited to file your observations and insofar as the deficiencies are such as to be rectifiable, to correct the indicated deficiencies within a period

of 4 months

from the notification of this communication, this period being computed in accordance with Rules 78(2) and 83(2) and (4) EPC.

One set of amendments to the description, claims and drawings is to be filed within the said period on separate sheets (Rule 36(1) EPC).

Failure to comply with this invitation in due time will result in the application being deemed to be withdrawn (Article 96(3) EPC).



Goya, J
Primary Examiner
for the Examining Division

Enclosure(s): 3 page/s reasons (Form 2906)

**Bescheid/Protokoll (Anlage)**

Datum
Date 12.09.2005
Date

Communication/Minutes (Annex)

Blatt
Sheet 1
Feuille

Notification/Procès-verbal (Annexe)

Anmelde-Nr.
Application No. 03 009 717.4
Dessin n°

646

The examination is being carried out on the **following application documents**:

Description, Pages

1-3, 5-19, 21-36, 38, as originally filed
39

4, 4a, 4b, 20, 37, 40 received on 10.01.2005 with letter of 05.01.2005

Claims, Numbers

1-29 received on 10.01.2005 with letter of 05.01.2005

Drawings, Sheets

1/7-7/7 as originally filed

- 1 The following documents are referred to in this communication; the numbering will be adhered to in the rest of the procedure:

D1: US-A-5 898 870 (IWAKURA YOSHIYUKI ET AL) 27 April 1999
D2: US-A-6 078 955 (KONNO CHISATO ET AL) 20 June 2000

- 2 The amendments and arguments filed with the letter of 05 January 2004 have been considered. However, they fail to place the application in a condition for the grant of the patent.

- 2.1 Document D1, considered the closest prior art, discloses in terms of **claim 1**, in a



Datum
Date 12.09.2005
Date

Blatt
Sheet 2
Feuille

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

computing environment (abstract), a method comprising:

- at a first computing device, discovering a second computing device (column 3, lines 25-46);
- at the first device, obtaining data related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device (column 3, lines 25-46); and
- the first device interpreting the data to access a resource of the second device (column 3, lines 35-50;
- the first device performing a computing task including using the resource of the second device to perform at least part of the task (column 2, lines 17-25; column 6, lines 56-65; column 9, lines 18-25. A computer can include a list of "works" or processes and only send a part of them for execution to another computer).

Consequently, the subject-matter of claim 1 is not new in the sense of Article 54(1) and (2) EPC, and claim 1 is therefore not allowable under Article 52(1) EPC.

2.2 **Claim 7** is a representation of claim 1 in terms of system, and consequently, it is also not allowable due to lack of novelty in the sense of Article 54(1) and (2) EPC.

2.3 Document D2 discloses, in terms of **claim 17**, a computer-readable medium having stored thereon a resource information exchange data structure (column 4, lines 52-57; column 5, lines 62-63; figure 5), the resource information exchange data structure:

- a first field for identifying a type of resource (column 6, lines 9-14; figure 5. First column in table 502);
- a second field for identifying the second device that includes a resource corresponding to the type of resource identified in the first field (column 6, lines 1-5 and figure 5. First column in table 500);
- a set of at least one third field, each third field in the set corresponding to the second field for identifying information of the second device identified by the first field (column 6, lines 17-26; figure 5: Table 505).

Datum
Date
Date 12.09.2005Blatt
Sheet
Feuille 3Anmelde-Nr.
Application No.: 03 003 717.4
Demande n°:

The only difference between the subject-matter of claim 17 and the disclosure of document D2 is:

- the table being suitable for providing resource information of a second device to a first computing device.

The technical problem associated with the only difference is how to provide with the required information to a device so that it can take decisions about the place of execution of processes in a distributed system architecture. This technical problem is well known in the field of distributed architectures and has already been solved in the same way, see for example document D1 (column 3, lines 15-50. The "resource utilization-state management table" provides information to the first device about the second device and using such information the first device decides where to execute the processes).

Consequently, claim 17 does not involve an inventive step in the sense of Article 56 EPC, and is therefore not allowable under Article 52(1) EPC.

- 2.4 The dependent claims also do not meet the requirements of Article 52(1) EPC (the applicant is referred to the first communication).
- 3 It is not at present apparent which part of the application could serve as a basis for a new, allowable claim. Refusal of the application under Article 97(1) EPC is therefore to be expected.
- 4 The Request for Oral Proceedings has been noted. Should the applicant not overcome the raised objections, then summons for Oral Proceedings would probably be the next step in the Examination procedure.

648

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER
ANWALTSSOZIELTÄT

EPO - Munich
59

10. Jan. 2005

GKS & S. MAXIMILIANSSTRASSE 58 D-80538 MÜNCHEN GERMANY

EUROPEAN PATENT OFFICE
ERHARDTSTR. 27
80298 MÜNCHEN

RECHTSANWÄLTE
LAWYERS
MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRÖDER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANK, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA BERTRAM, LL.M.
KARIN LOCHNER
SABETT ERTLE
CHRISTINE NEUMERL
SABINE PRÜCKNER
CORNELIA SCHMITT
BERNHARD MEHNERT, DIPL. ING.
PETRA LÜBBE

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS
MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOS
HANS HILGERS
DR. HENNING MEYER-PLATH
ANNELIE EHNGOLD
THOMAS SCHUSTER
DR. KLARA GOLDBACH
MARTIN AUFENANGER
GOTTFRIED KLITZSCH
DR. HEIKE VOGEL-SANG-WENKE
REINHARD KNAUER
DIETMAR KUHL
DR. FRANZ JOSEF ZIMMER
BETTINA C. REICHELT
DR. ANTON K. PFÄU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (U.S.P.A.) M.S.
BERND ROTHMANN
DR. DANIELA KINKELDEY
THOMAS VV. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS EICKELKAMP
JOCHEN KILCHERT
DR. ANJA DIETEL

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS
BERLIN
PROF. DR. MANFRED BÖNING
DR. PATRICK ERK, M.S. (U.S.P.A.)
KÖLN
DR. MARTIN DROPMANN
CHEMNITZ
DR. CARVEN STEINIGER
PATENTANWÄLTE
ALICANTE
WOLFGANG MEISTER
DR. ANDREAS TORKA
RECHTSANWÄLT
DR. WILFRIED STOCKMAIR
(ISSG)

IHR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.

EP27314TE109ism

DATUM / DATE

05.01.2005

Application No: 03 009 717.4-1244
Applicant: MICROSOFT CORPORATION

In response to the Official Communication dated September 2, 2004:

Submitted herewith are:

new claims 1 to 29 to replace all claims presently on file; and

new description pages 4, 4a, 4b, 20, 37 and 40 to replace pages 4, 20, 37 and 40 presently on file.

New claim 1 is based on previous claims 1 and 7. New claims 2 to 5 correspond to previous claims 2 to 5 respectively. New claim 6 is based on previous claims 6 and 7. New claim 7 is based on previous claim 7. New claims 8 to 16 correspond to previous claims 8 to 16 respectively. New claim 17 is based on previous claim 17 and description page 22, lines 8 to 10. New claims 18 to 29 correspond to previous claims 18 to 29 respectively.

The claims have been clarified and have been supplemented with reference signs.

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
<http://www.grunecker.de>
e-mail: info@grunecker.de

DEUTSCHE BANK MÜNCHEN
AG. 17 51734
BLZ 700 700 10
SWIFT: DEUT DE 33

317

The description has been supplemented with an indication of the prior art and has been adapted to the claims. Further, statements on pages 20, 37 and 40 have been cancelled to overcome the objections made in sections 5.1 and 5.2 of the Official Communication.

The clarity objections raised in section 4 of the Official Communication are believed to be overcome. In detail, the term "first" in independent claim 7 and its dependent claims has been deleted, and the claim reference of dependent claims 14 and 15 has been changed from claim 7 to claim 13.

Turning now to the novelty objections made in section 2 of the Official Communication, it is respectfully disagreed.

In detail, document D1 relates to load sharing in a parallel computer system having a computer group constituted by a plurality of computers, and an operation management mechanism which is a computer for managing the operation of the computer group (column 2, lines 61 to 65). Firstly, resource utilization target values by work are set in advance in the operation management mechanism, and subsequently the respective computers of the computer group inform the operation management mechanism of collected resource utilization states by work. In response to a request by a computer to newly start a work process, the resource utilization target value of the work process is compared with the resource utilization state thereof on a resource utilization state management table which is prepared in the initialization of the system at the operation management mechanism. Further, one of the computers is selected from the plurality of computers constituting the computer group on the basis of the result of the comparison as the optimum system and is prompted to execute the work process (see column 3, lines 34 to 46 and column 9, lines 3 and 4).

Document D2 relates to a method of controlling a computer system including a plurality of computers interconnected by a network and used by a plurality of users. The computer system includes at least a server, an administration server and a plurality of terminals. The administration server is responsible for managing all the computer resources and all the jobs performed on the computers as well as the entire user information in the computer system (column 4, lines 22 to 31). The administration server collects and manages information in a system resource management table. Further, when a user logs in to the system at a terminal, the log-in information and the terminal information used for the log-in are sent to

the administration server (column 6, lines 60 to 63). After a processing request is sent from a terminal to the administration server, a server or terminal judged to be capable of handling a given process most efficiently is determined from the prevailing processing load of the servers in the network and the property and the location of the file and the program used for the intended process, and the place of execution of the process is notified to the particular terminal. The terminal then requests a computer corresponding with the notification to execute the process (column 4, lines 32 to 45).

It is respectfully submitted that the subject matter of new claims 1, 6 and 7 differs from the disclosure of D1 and D2 at least in that a computing device is described which performs a computing task including a resource of another device to perform at least part of the task. Thus, the subject matter of claims 1, 6 and 7 is novel in view of documents D1 and D2.

Referring to new claim 17, it is respectfully submitted that the described data structure for exchanging resource information is not disclosed in document D1 or D2. Those documents describe resource management tables. In particular, D2 discloses several tables which together form a system resource management table for collecting and managing information regarding the system's resources. This table structure differs from a formatted message including resource information which can be sent to a device as described in the present application. Thus, the subject matter of claim 17 is novel over documents D1 and D2.

Discussing the inventive step of the independent claims and starting from document D1, the object of the invention is a better performance of computing tasks in a computing environment containing various computing devices (see also description page 3, line 23 to page 4, line 3).

The skilled person trying to solve the object of the invention is taught by document D1 to use a parallel computer system having a computer group constituted by a plurality of computers and one operation management mechanism which is a computer for managing the operation of the computer group. The operation management mechanism compares resource utilization target values with the resource utilization states of the computers constituting the computer group to select one of the computers to execute a newly-requested work process. The compared values could be for example the CPU utilization ratio or the state of the free main storage.

Assuming that the skilled person takes the disclosure of document D2 into account, the skilled person might add terminal computers to the parallel computer system for users to log into the system. D2 also suggests comparing the CPU processing performance, the memory capacity and additionally the file capacity to determine the place of execution of a requested process (see D2, column 9, lines 20, 21, 41 and 42).

In conclusion, both documents teach the skilled person to select one out of the plurality of computers to perform a requested task. This is not what is taught by the present application defining a system where a computing task is performed on a device including using a resource of another device to perform at least part of the task. Thus, it is not rendered obvious by document D1, D2 or a combination thereof to perform a computing task in the manner as claimed. It is respectfully submitted therefore that the subject matter of the independent claims involves an inventive step.

As the application therefore now appears to be in good condition for grant of a patent, the Examining Division is respectfully requested to issue a Communication according to Rule 51(4) EPC as the next step of the proceedings. As a matter of precaution only, oral proceedings are requested.


(Eickelkamp)

Encls.: New claims 1 to 29
 New description pages 4, 4a, 4b, 20, 37 and 40

US-A-5 898 870 describes load sharing in a parallel computer system having a computer group constituted by a plurality of computers, and an operation management mechanism which is a computer for managing the operation of the computer group. Firstly, resource utilization target values by work are set in advance in the operation management mechanism, and subsequently the respective computers of the computer group inform the operation management mechanism of collected resource utilization states by work. In response to a request by a computer to newly start a work process, the resource utilization target value of the work process is compared with the resource utilization state thereof on a resource utilization state management table which is prepared in the initialization of the system at the operation management mechanism. Further, one of the computers is selected from the plurality of computers constituting the computer group on the basis of the result of the comparison as the optimum system and is prompted to execute the work process.

US-A-6 078 955 describes a method of controlling a computer system including a plurality of computers interconnected by a network and used by a plurality of users. The computer system includes at least a server, an administration server and a plurality of terminals. The administration server is responsible for managing all the computer resources and all the jobs performed on the computers as well as the entire user information in the computer system. The administration server collects and manages information in a system resource management table. Further, when a user logs in to the system at a terminal, the log-in information and the terminal information used for the log-in are sent to the administration server. After a processing request is sent from a terminal to the administration server, a server or terminal judged to be capable of handling a given process most efficiently is determined from the prevailing processing load of the servers in the network and the property and the location of the file and the program used for the intended process, and the place of execution of the process is notified to the particular terminal. The terminal then requests a computer corresponding with the notification to execute the process.

SUMMARY OF THE INVENTION

It is the object of the invention to have a better performance of computing tasks in a computing environment containing various computing devices.

This object is solved by the invention as claimed in the independent claims.

Preferred embodiments are defined by the dependent claims.

needed is a common, consistent way to remotely discover and expose computing resources of various computing devices to one another for cross-resource usage.

5

SUMMARY OF THE INVENTION

~~Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.~~

10

15 Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus

20

~~abstracted into its resources, which are then programmatically~~

~~needed is a common, consistent way to remotely discover and
expose computing resources of various computing devices to one
another for cross-resource usage.~~

5

~~SUMMARY OF THE INVENTION~~

Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.

10 Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus abstracted into its resources, which are then programmatically

20

and/or the device identity. Further note that the security mechanism may operate per identity and per resource, e.g., one entity may be authorized to use a device's display, but not its storage, while another entity may be able to use any of that same device's available resources, including its storage. This is similar to the .NET security model in which services provide methods that control access to a users' data based on the requesting user's identity, defined role and scope for that role. In this way, data can be accessed by its owner, and shared to an extent determined by the owner. In the present invention, instead of data it is a device resource being accessed, whereby a user's identification, defined role and scope for that role determine the extent to which a resource may be accessed.

Indeed, the security mechanisms of the present invention are based on those essentially those employed for security / access control in .NET services, as generally described in United States patent applications, serial nos. 10/003,754 entitled "IDENTITY-CENTRIC DATA ACCESS" and serial number 10/003,767, entitled "AUTHORIZING A REQUESTING ENTITY TO OPERATE UPON DATA STRUCTURES" filed on October 22, 2001, assigned to the assignee of the present application, ~~and herein incorporated by reference.~~ To this end, at least a thin subset of the .NET service runs on each device which

an input resource the on computer 702, storage resource on the computer 703, and an output resource on the computer 704. The computer 704 acts as a coordinating authority for the implementation 700. Although the resources are on different
5 computer systems, the resources are accessed in accordance with the present invention to essentially form a single functioning system.

The shell resource 710 calls the application 712, and also connects to the other devices' resources through ports
10 724, 726 and 728, conceptually represented in FIG. 7 as one port per resource, although a separate input and output port may be provided. Ports essentially comprise structured queues that can be written (output) to or read (input) from via a handle to that port. Ports are mobile, data is not lost, and
15 security can be associated with a port such that no one except the intended entity will see the data intended for it.

The shell 710 used in the implementation 700 is based on a programming language referred to as xSpresso, as described in United States provisional patent application serial number
20 60/379,864, filed on May 10, 2002, ~~and herein incorporated by reference in its entirety.~~ In general, xSpresso is directed to a programming language and runtime that brokers control and data flows among asynchronous system components. Among other things, xSpresso expresses the cooperative dimensions of

numerous benefits to a user and generally improves the user's computing experience. The method and system are schema-based, and are extensible and highly flexible.

While the invention is susceptible to various
5 modifications and alternative constructions, certain
illustrated embodiments thereof are shown in the drawings and
have been described above in detail. It should be understood,
however, that there is no intention to limit the invention to
the specific forms disclosed, but on the contrary, the
10 intention is to cover all modifications, alternative
constructions, and equivalents falling within the ~~spirit and~~
scope of the invention.

Application No.: 03 009 717.4

Applicant: MICROSOFT CORPORATION

Our Ref.: EP27314TE109Ism

Date: January 5, 2005

NEW CLAIMS

1. A method in a computing environment, the method comprising:

at a first computing device (202, 400, 500), discovering a second computing device (212, 402, 404, 502, 504);

at the first device, obtaining data (209A, 209B) related to the second device, the data including resource information of the second device formatted according to a schema understood by the first device;

the first device interpreting the data to access a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) of the second device; and

the first device performing a computing task including using the resource of the second device to perform at least part of the task.

2. The method of claim 1 further comprising:

communicating (600) a request to use the resource from the first device to the second device.

3. The method of claim 1 or 2 further comprising:

communicating identity information from the first device to the second device.

4. The method of claim 3 wherein the identity information corresponds to credentials (207A, 207B).
5. The method of claim 4 wherein the credentials are provided by an authentication service (220).
6. A computer-readable medium having computer-executable instructions to perform a method, the method comprising:

at a first computing device (202, 400, 500):

locating information (209A, 209B) about a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) of at least one other device (212, 402, 404, 502, 504) that is available for use, the information located based on an identity associated with the first computing device;

selecting a resource of a second device from the information;

accessing the resource of the second device; and

performing a computing task including using the resource of the second device to perform at least part of the task.

7. A system in a computing environment, the system comprising:
 - a first computing device (212, 402, 404, 502, 504) associated with at least one identity, the first device including a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) and configured to provide access to the resource by at least one other device (202, 400, 500) associated with the identity, the first computing device providing information (209A, 209B) about the resource according to a resource schema associated with the at least one identity;
 - a second device, the second device configured to request access to the resource; and

- the second device performing a computing task including using the resource on the first device to perform at least part of the task.
8. The system of claim 7 wherein the resource comprises storage (141, 190, 716).
 9. The system of claim 7 wherein the resource comprises bandwidth of a connection (171, 173) from the first device to at least one other computing device.
 10. The system of claim 7 wherein the resource comprises processing cycles of a processor.
 11. The system of claim 7 wherein the resource comprises an input method.
 12. The system of claim 7 wherein the resource comprises an output method.
 13. The system of one of claims 7 to 12 further comprising a third computing device connected to the first computing device, wherein the second device accesses the resource to perform an operation with the third computing device.
 14. The system of claim 13 wherein the third computing device comprises a printer (402), and wherein the resource corresponds to a printer driver (406) on the first computing device.
 15. The system of claim 13 wherein the third computing device comprises a server (502), and wherein the first computing device acts as a proxy to the server for the second computing device.
 16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.
 17. A computer-readable medium having stored thereon a resource information exchange data structure being suitable for providing resource information

(209A, 209B) of a second device to a first computing device, the resource information exchange data structure comprising:

a first field for identifying a type of resource;

a second field for identifying the second device that includes a resource (R_{A1} - R_{Am} , R_{B1} - R_{Bn}) corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying the resource information of the second device identified by the first field.

18. The computer-readable medium of claim 17 wherein the first field includes information corresponding to a storage resource type.
19. The computer-readable medium of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
20. The computer-readable medium of claim 18 or 19 wherein the third field set includes information corresponding to free space data of the storage resource.
21. The computer-readable medium of claim 17 wherein the first field includes information corresponding to bandwidth data.
22. The computer-readable medium of claim 17 wherein the wherein the first field corresponds to processing power.
23. The computer-readable medium of claim 22 wherein the third field set includes information corresponding to speed data of a processor.
24. The computer-readable medium of claim 22 or 23 wherein the third field set includes information corresponding to usage data of a processor.

25. The computer-readable medium of one of claims 22 to 24 wherein the third field set includes information corresponding to size data of a memory.
26. The computer-readable medium of claim 25 wherein the third field set includes information corresponding to usage data of a memory.
27. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a personal computer (404).
28. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a disk drive (141, 151, 155).
29. The computer-readable medium of one of claims 17 to 26 wherein the second field includes information identifying the device as a telephone (400).

658



EPA/EPO/OEB
D-80298 München
+49 89 2399-0
TX 523 656 epmu d
FAX +49 89 2399-4465

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Generaldirektion 2

Directorate General 2

Direction Générale 2

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Telephone numbers: Branch at The Hague

Primary Examiner (substantive examination) -31 70 340-2370

Formalities Officer / Assistant (Formalities and other matters) -31 70 340-2346



Application No.
03 009 717.4 - 1244

Ref.
EP27314-034/df

Date
02.09.2004

Applicant
MICROSOFT CORPORATION

Communication pursuant to Article 96(2) EPC

The examination of the above-identified application has revealed that it does not meet the requirements of the European Patent Convention for the reasons enclosed herewith. If the deficiencies indicated are not rectified the application may be refused pursuant to Article 97(1) EPC.

You are invited to file your observations and insofar as the deficiencies are such as to be rectifiable, to correct the indicated deficiencies within a period

of 4 months

from the notification of this communication, this period being computed in accordance with Rules 79(2) and 83(2) and (4) EPC.

One set of amendments to the description, claims and drawings is to be filed within the said period on separate sheets (Rule 36(1) EPC).

Failure to comply with this invitation in due time will result in the application being deemed to be withdrawn (Article 96(3) EPC).



GOYA ALONSO J
Primary Examiner
for the Examining Division

Enclosure(s): 5 page/s reasons (Form 2906)

332



Bescheid/Protokoll (Anlage)

Communication/Minutes (Annex)

Notification/Procès-verbal (Annexe)

Datum
Date 02.09.2004
Date

Blatt
Sheet 1
Feuille

Anmelde-Nr.
Application No.: 03 009 717.4
Demande n°

The examination is being carried out on the **following application documents**:

Description, Pages

1-40 as originally filed

Claims, Numbers

1-29 as originally filed

Drawings, Sheets

1/7-7/7 as originally filed

- 1 The following documents are referred to in this communication; the numbering will be adhered to in the rest of the procedure:

- D1: US-A-5 898 870 (IWAKURA YOSHIYUKI ET AL) 27 April 1999 (1999-04-27)
- D2: US-A-6 078 955 (KONNO CHISATO ET AL) 20 June 2000 (2000-06-20)
- D3: US-A-5 640 569 (HIBBERT CHRISTOPHER T ET AL) 17 June 1997 (1997-06-17)
- D4: US-B1-6 247 026 (WALDO JAMES H ET AL) 12 June 2001 (2001-06-12)
- D5: EP-A-0 952 513 (HEWLETT PACKARD CO) 27 October 1999 (1999-10-27)
- D6: WO 00 51031 A (AMERICA ONLINE INC ;HENDREN C HUDSON III (US)) 31 August 2000 (2000-08-31)

- 2 The present application does not meet the requirements of Article 52(1) EPC because the subject-matter of the independent **claims 1, 6, 7 and 17** is not new in the sense of Article 54(1) and (2) EPC.

- 2.1 Document D1, which is considered the closest prior art, discloses in terms of



660

claim 1, a method comprising:

- at the first device (one of the *respective computers*, line 30 in col. 3; *any computer*, line 34 in col. 3; *the request issuing computer*, line 38 in col. 3), obtaining data related to the second device (*selected computer*, line 46 in col. 3), the data including resource information of the second device formatted according to a schema understood by the first device (col. 3, lines 25-34).
- the first device interpreting the data to access a resource of the second device (col. 3, lines 34-50: *the request issuing computer* performs a comparison using the resource information of the *resource utilisation-state management table*, and *selects* a second computer. After this selection, the first computer *instructs the selected computer to execute the work process*).

Moreover, a necessary implication of the fact that the first computing device accesses a resource of the second computing device, is that the first computing device discovers the second computing device, thus the remaining step specified in claim 1 is also present in D1: first computing device discovering a second computing device.

Consequently, D1 discloses all the features of claim 1.

2.2 Document D1 discloses in terms of **claim 6**, a computer-readable medium having executable instructions to perform a method (col. 9, lines 25-31. Since this program runs on a computer device, it needs to be stored locally on some appropriate storage medium), the method comprising:

- locating information about a resource of at least one other device that is available for use, the information located based on an identity associated with the first computing device (col. 3, lines 25-34 and figure 2, the identity being "*System 1*").
- selecting a resource of a second device from the information (col. 3, lines 42-45. The resource being the processing capacity of the selected computer).
- accessing the resource of the second device (col. 3, lines 46-50):

2.3 **Claim 7** is a representation in terms of system of the computer-readable medium comprising the executable instructions of claim 6. Document D1 discloses a

334

Datum:
Date: 02.09.2004
Date:Blatt:
Sheet: 3
Feuille:Anmelde-Nr.:
Application No.: 03 009 717.4
Demande n°:

computer environment (abstract) that implements the instructions of claim 6, and therefore, claim 7 is not allowable due to lack of novelty, Article 52(1) and Article 54(1) and (2) EPC.

2.4 Document D2 discloses in terms of **claim 17**, a computer-readable medium having stored thereon a data structure (col. 4, lines 52-57; col. 5, lines 62-63, and figure 5), comprising:

- a first field for identifying a type of resource (col. 6, lines 9-14, and figure 5: First column in table 502).
- a second field for identifying a device that includes a resource corresponding to the type of resource identified in the first field (col. 6, lines 1-5 and figure 5: First column in table 500).
- a set of at least one third field, each third field in the set corresponding to the second field for identifying information about the device resource identified by the first field (col. 6, lines 17-26, and figure 5: Table 505).
- data structure is used to provide resource information to another device (*administration server*) that is different from the one identified in the second field (*particular server*) such that the other device can use a resource identified in the third field set (col. 5, lines 32-39).

3 The following is noted regarding the dependent claims:

3.1 The following claims lack novelty because their additional features are known from the above cited documents D1 and D2:

- **Claims 2:** Document D1, col. 3, lines 46-50.
- **Claims 18-20 and 22-27:** Document D2, abstract; col. 6, lines 17-26, and figure 5: Tables 500, 501, 502 and 505.

3.2 The following claims lack an inventive step because their features are considered common measures or well-known implementation-related design options for a skilled person:

- **Claims 3-5:** See for example, document D4 discloses (claims 11, 12, 15 and 26) the use of credentials and authentication servers for the authentication of the access from one computer to another in order to share resources.



Datum
Date 02.09.2004
Date

Blatt
Sheet 4
Feuille

Anmelde-Nr.
Application No. 03 009 717.4
Demande n°

- **Claims 8, 10, 12:** See for example document D2, abstract; col. 18, lines 30-37; col. 6, lines 1-5 and figure 5.
 - **Claim 13:** See for example document D2, col. 17, lines 10-25.
 - **Claims 14 and 15:** See for example document D5 for the additional feature of claim 14 (claim 1 and figure 1; *printer server* has the driver of a printer) and document D6 for the additional feature of claim 15 (page 2, lines 18-21; *proxy*).
 - **Claims 11 and 16:** The additional features of these claims are considered common measures for a person skilled in the art.
 - **Claim 9 and 21:** See for example document D3 that discloses (col. 3, lines 31-35) the allocation of bandwidth as a computer resource.
 - **Claims 28-29:** The additional features of these claims are considered common measures for a person skilled in the field of computer resources monitoring and allocation.
- 4 The present application does not meet the requirements of Article 84 in respect to clarity and conciseness:
- 4.1 The term "third computing device" in **claims 14 and 15** lacks an antecedent. This term is only defined in claim 13, and it is, therefore, considered that claims 14 and 15 are probably meant to be dependent claims on claim 13, and not claim 7. Clarification is required.
- 4.2 Although independent claim 7 and the depending claims on it (claims 8-13) specify a "first" resource, they never define a "second" resource. The word "first" in connection to "resource" should therefore be removed.
- 5 Applicant is requested to also attend the following matters:
- 5.1 On page 20, line 23, and on page 37, lines 20-21, a statement is made with regard to reference documents, which are to be incorporated in the application by reference. Since these documents appear not to be essential to understand and carry out the invention of the instant application, the incorporation of these documents is not necessary. Consequently, the statement "incorporated by reference" has to be removed (Guidelines, C-II, 4.18).
- 5.2 The vague and imprecise statement at the end of the description on page 40

Datum
Date 02.09.2004
DateBlatt
Sheet 5
FeuilleAnmeld.-Nr.
Application No. 03 009 717.4
Demande n°

implies that the subject-matter for which protection is sought may be different to that defined by the claims, thereby resulting in lack of clarity of the claims (Article 84 EPC) when used to interpret them (see the Guidelines, C-III, 4.3a). The sentence, from line 7 to line 12 should therefore be removed.

- 5.3 To meet the requirements of Rule 27(1)(b) EPC, the documents D1 and D2 should be identified in the description and the relevant background art disclosed therein should be briefly discussed.
- 5.4 Amended claims should follow the two-part form in accordance with Rule 29(1) EPC, with those features known in combination from the prior art (D1) being placed in the preamble (Rule 29(1)(a) EPC) and with the remaining features being included in the characterising part (Rule 29(1)(b) EPC).
- 5.5 When filing amended claims the applicant should at the same time bring the description into conformity with the amended claims (Rule 27(1)(c) EPC).
- 5.6 The features of the claims should be provided with reference signs placed in parentheses to increase the intelligibility of the claims (Rule 29(7) EPC). This applies to both the preamble and characterising portion (see the Guidelines, C-III, 4.11).
- 5.7 In order to facilitate the examination of the conformity of the amended application with the requirements of Article 123(2) EPC, the applicant is requested to clearly identify the amendments carried out, and to indicate the passages of the application as filed on which these amendments are based (Guidelines, E-II, 1).
- 5.8 Moreover, when drafting a new set of claims, the requirements of unity, Article 82, should be met, i.e. all independent claims should be linked though a single general inventive concept (Rule 30 EPC).

J. GOYA

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER

ANWALTSSOZ. ETÄT

EPO - Munich
39

07. Juli 2004

GKS & S. MAXIMILIANSTRASSE 58 D-80538 MÜNCHEN GERMANY

EUROPEAN PATENT OFFICE

ERHARDTSTR. 27
80298 MUENCHEN

RECHTSANWÄLTE
LAWYERS

MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRODER, LL.M.
CHRISTA NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANK, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA PFRANG, LL.M.
KARIN LOCHNER
BABETT ERTL
CHRISTINE NEUBERGER
SABINE PRÜCKNER

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER H. JAKOB
WOLFGANG MEISTER
HANS PILGERS
DR. HENNING MEYER-PLATH
ANNELE EHNOLD
THOMAS SCHJETER
DR. KLARA GOLDBACH
MARTIN AUFENANGER
GOTTFRIED KLITZSCH
DR. HEIKE VOGELSONG WENKE
REINHARD KNAUER
DIETMAR KUPF
DR. JOHANN JOSEF ZIMMER
SETTINA K. REICHELT
DR. ANTON K. PRALL
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (U.G.P.A.) M.S.
BERND ROTHMAYER
DR. DANIELA KINKELDEY
THOMAS W. LAURENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS EICKELKAMP
JOCHEN KILCHERT
DR. THOMAS FRIED
DR. ANJA DIETEL

PATENTANWÄLTE
EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED BÖNING
DR. PATRICK ERK, M.S. (M.P.)
KÖLN
DR. MARTIN GROPMANN
CHEMNITZ
MANFRED SCHNEIDER

OF COUNSEL
PATENTANWÄLTE
DR. GÜNTER BEZOLD

DR. WILFRIED STOCKMAIR
(1996)

IHR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.

DATE / DATE

EP27314NN904awe

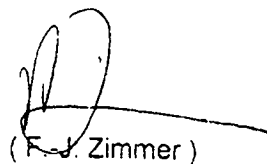
07.07.04

European Patent Application No.: 03 009 717.4
Applicant: MICROSOFT CORPORATION

The following official fees are herewith paid with the enclosed EPO Form 1010:

- Examination Fee
- Designation Fee for all contracting states (=7)

Please debit our account no. 28 00 04 37 with the total amount of EUR 1.955,--.


(F. J. Zimmer)

Zur Kasse

Encl.
-EPO Form 1010

GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
http://www.gruenecker.de
e-mail: info@gruenecker.de

DEUTSCHE BANK MÜNCHEN
No. 17 51734
BLZ 700 700 10
SWIFT: DEUTDE33



P.B. 5818 - Patentamt
2280 HV Rijswijk (Z-H)
T (070) 3 40 20 40
FAX (070) 3 40 30 10

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Gesamtdirektion :

Directorate General :

Direction générale :

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



epoline® Customer Services

Tel.: +31 (0)70 340 45 00

Date
19.01.04

Reference EP27314-034/df	Application No./Patent No. 03009717.4 - 2211
Applicant/Proprietor MICROSOFT CORPORATION	

**Communication pursuant to Rule 50 EPC - reminder of payment of the designation fees
(Art. 79(2) EPC) and of the examination fee (Art. 94(2) EPC)**

The date on which the European Patent Bulletin mentions the publication of the European search report for the above-mentioned European patent application is: 14.01.04.

Your attention is drawn to Article 79(2) and (3) EPC as well as Article 94(2) and (3) EPC according to which within **SIX MONTHS** after the above-mentioned publication date of the search report

- the designation fee(s) must be paid,
- a written request for examination must be filed as well as the examination fee must be paid.
(A written request for examination has been filed already.)

The current rate of the designation fee for each contracting state designated is: EUR 75,00

If the application has been filed on or after 01 July 1999 the payment of seven times the amount of the designation fee is deemed to constitute payment of the designation fees for all contracting states (see OJ EPO 06/1999, 405).

The current rate of the examination fee is: EUR 1430,00

If at least one designation fee and the examination fee are not paid within the period laid down in Article 79(2) or 94(2) EPC, the application shall be deemed to be withdrawn (Arts. 79(3), 94(3) EPC).

Any extension fees are also payable within the above-mentioned period.

666



Date

Sheet 2

Application No. 03009717.4

NOTE TO USERS OF THE AUTOMATIC DEBITING PROCEDURE:

1) Designation fees

If the application has been filed up to 30 June 1999, the designation fees for the contracting states marked under no. 2 of section 32 of the Request for Grant (EPO Form 1001) will be debited on the last day of the period pursuant to Article 79(2) EPC, unless the EPO receives prior instructions to the contrary.

If the application has been filed on or after 01 July 1999, seven times the amount of the designation fee will be debited on the last day of the period pursuant to Article 79(2) EPC. However, if contracting states are marked under no. 2 of section 32 of the Request for Grant (EPO Form 1001), the designation fees only for these contracting states will be debited unless instructions to the contrary have reached the EPO within the basic period for paying the designation fees.

2) Examination fee

Unless the EPO receives prior instructions to the contrary, the examination fee will be debited on the last day of the period for payment.

For further details see the Arrangements for the automatic debiting procedure, Supplement to OJ EPO 02/2002.

RECEIVING SECTION



340

667



P B 5616 - Patentlaan 2
2250 HV Rijswijk (ZH)
☎ +31 70 340 2040
TX 31651 epo nl
FAX +31 70 340 3016

Europäisches
Patentamt

Eingangs-
stelle

European
Patent Office

Receiving
Section

Office européen
des brevets

Secteur de
Dépôt

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Datum/Date

09/01/04

Zeichen/Ref./Réf. EP27314-034/df	Anmeldung Nr./Application No./Demande n°/Patent Nr./Patent No./Brevet n° 03009717.4-2211/1376977
Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire MICROSOFT CORPORATION	

MITTEILUNG GEMÄSS TEIL A-III, 5.3 DER RICHTLINIEN FÜR DIE PRÜFUNG IM EPA

Die dem nachstehend angegebenen Erfinder gemäss Regel 17(3) EPÜ über-
sande Mitteilung kam unzustellbar zurück. Bitte teilen Sie uns die
richtige Adresse des Erfinders mit (Regel 17(1) EPÜ).

NOTIFICATION PURSUANT TO PART A-III, 5.3 OF THE GUIDELINES FOR
EXAMINATION IN THE EPO

The communication issued pursuant to Rule 17(3) EPC, sent to the inven-
tor designated below, has been returned by the postal services. You are
requested to indicate the correct address of the inventor (Rule 17(1)).

NOTIFICATION FAITE EN APPLICATION DE LA PARTIE A-III, 5.3 DES DIRECTIVES
RELATIVES A L'EXAMEN PRATIQUE A L'OEB

La communication selon la règle 17(3) CBE, transmise à l'inventeur
désigné ci-après, a été retournée par les services postaux. Vous êtes
prié d'indiquer l'adresse exacte de l'inventeur (règle 17(1) CBE).

ERFINDER : Alles, David
INVENTOR : 4123 Evanston Avenue North
INVENTEUR : US / Seattle, Washington 98103

EINGANGSSTELLE
RECEIVING SECTION
SECTION DE DEPOT





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 00 9717

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 5 898 870 A (IWAKURA YOSHIYUKI ET AL) 27 April 1999 (1999-04-27) * abstract * * column 3, line 25-50 * * column 9, line 25-31 * * figure 2 *	1-16	G06F9/46 H04L29/06
X	US 6 078 955 A (KONNO CHISATO ET AL) 20 June 2000 (2000-06-20) * column 4, line 52-57 * * column 5, line 32-39 * * column 5, line 62,63 * * column 6, line 1-26 * * figure 5 *	17-29	
A	US 5 640 569 A (HIBBERT CHRISTOPHER T ET AL) 17 June 1997 (1997-06-17) * column 3, line 31-35 *	9,21	
A	US 6 247 026 B1 (WALDO JAMES H ET AL) 12 June 2001 (2001-06-12) * claims 11, 12, 15 and 26 *	3-5	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
A	EP 0 952 513 A (HEWLETT PACKARD CO) 27 October 1999 (1999-10-27) * claim 1 * * figure 1 *	14	H04L G06F
A	WO 00 51031 A (AMERICA ONLINE INC ;HENDREN C HUDSON III (US)) 31 August 2000 (2000-08-31) * page 2, line 18-21 *	15	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 21 November 2003	Examiner Goya, J
CATEGORY OF CITED DOCUMENTS			
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

4

EP03-CPM 15/6 10.02 (P04C01)

342

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.

EP 03 00 9717

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on:
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information

21-11-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5898870 A	27-04-1999	JP 9167141 A	24-06-1997
US 6078955 A	20-06-2000	JP 10074159 A	17-03-1998
US 5640569 A	17-06-1997	NONE	
US 6247026 B1	12-06-2001	US 5832529 A	03-11-1998
		AU 3297299 A	15-09-1999
		CN 1298509 T	06-06-2001
		EP 1062580 A1	27-12-2000
		JP 2002505470 T	19-02-2002
		WO 9944130 A1	02-09-1999
		EP 0836140 A2	15-04-1998
		JP 11045187 A	16-02-1999
		US 6016500 A	18-01-2000
		US 6263350 B1	17-07-2001
		US 2002116383 A1	22-08-2002
		US 2002138662 A1	26-09-2002
		US 6081813 A	27-06-2000
		US 2003093505 A1	15-05-2003
		US 6237009 B1	22-05-2001
		US 2003115374 A1	19-06-2003
		US 6327596 B1	04-12-2001
		US 6243716 B1	05-06-2001
		US 2003208512 A1	06-11-2003
		US 6519615 B1	11-02-2003
		US 6449648 B1	10-09-2002
		US 2001000812 A1	03-05-2001
		US 2002002576 A1	03-01-2002
		US 2002049779 A1	25-04-2002
		AU 2680299 A	15-09-1999
		AU 2680399 A	15-09-1999
		AU 2680499 A	15-09-1999
		AU 2686699 A	15-09-1999
		AU 2686799 A	15-09-1999
		AU 2766199 A	15-09-1999
		AU 2769899 A	15-09-1999
		AU 2770199 A	15-09-1999
		AU 2770299 A	15-09-1999
		AU 2770399 A	15-09-1999
		AU 2770499 A	15-09-1999
		AU 2770599 A	15-09-1999
		AU 2787699 A	15-09-1999
		AU 2787799 A	15-09-1999
		AU 2787899 A	15-09-1999
		AU 2876899 A	15-09-1999

EP0 FORM P1045

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

670

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.

EP 03 00 9717

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-11-2003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6247026 B1		AU 2876999 A	15-09-1999
		AU 2878399 A	15-09-1999
		AU 2878499 A	15-09-1999
		AU 3297199 A	15-09-1999
		AU 3300499 A	15-09-1999
		AU 3300599 A	15-09-1999
		AU 3309199 A	15-09-1999
		CN 1292115 T	18-04-2001
		CN 1292116 T	18-04-2001
		CN 1292113 T	18-04-2001
EP 0952513 A	27-10-1999	US 6301012 B1	09-10-2001
		EP 0952513 A1	27-10-1999
		JP 2000033755 A	02-02-2000
WO 0051031 A	31-08-2000	AT 228257 T	15-12-2002
		AU 3501000 A	14-09-2000
		CA 2363571 A1	31-08-2000
		DE 60000822 D1	02-01-2003
		DE 60000822 T2	10-07-2003
		DK 1157344 T3	17-03-2003
		EP 1157344 A1	28-11-2001
		ES 2189743 T3	16-07-2003
		JP 2002538525 A	12-11-2002
		PT 1157344 T	30-04-2003
		WO 0051031 A1	31-08-2000

EPO/EP/PO/EA

For more details about this annex, see Official Journal of the European Patent Office, No. 12/82

344

671



P.B. 5818 - Patentlaan 2
2260 HV Rijswijk (ZH)
☎ +31 70 340 2040
TX 31651 epo nl
FAX +31 70 340 3016

**Europäisches
Patentamt**

Zweigstelle
in Den Haag
Recherchen-
abteilung

**European
Patent Office**

Branch at
The Hague
Search
division

**Office européen
des brevets**

Département a
La Haye
Division de la
recherche

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Datum/Date

02.12.03

Zeichen/Ref./Réf.

EP27314-034/df

Anmeldung Nr./Application No./Demande n°./Patent Nr./Patent No./Brevet n°.

03009717.4-1244-

Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire

MICROSOFT CORPORATION

COMMUNICATION

The European Patent Office herewith transmits as an enclosure the European search report for the above-mentioned European patent application.

If applicable, copies of the documents cited in the European search report are attached.

☒ Additional set(s) of copies of the documents cited in the European search report is (are) enclosed as well.

The following specifications given by the applicant have been approved by the Search Division:

☒ abstract

☒ title

☐ The abstract was modified by the Search Division and the definitive text is attached to this communication.

The following figure will be published together with the abstract: 1A

REFUND OF THE SEARCH FEE

If applicable under Article 10 Rules relating to fees, a separate communication from the Receiving Section on the refund of the search fee will be sent later.



345

672



P.B. 5816 - Patentlaan 2
2260 HV Rijswijk (ZH)
☎ +31 70 340 2040
TX 3:65: epo.nl
FAX +31 70 340 3016

Europäisches
Patentamt

Eingangs-
stelle

European
Patent Office

Receiving
Section

Office européen
des brevets

Section de
Dépôt

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE

Datum/Date
19/11/03

Zeichen/Ref./Réf. EP27314-034/df	Anmeldung Nr./Application No./Demande n°/Patent Nr./Patent No./Brevet n° 03009717.4-1244 / 1376977
Anmelder/Applicant/Demandeur/Patentinhaber/Propriétaire/Titulaire MICROSOFT CORPORATION	

NOTIFICATION OF EUROPEAN PUBLICATION NUMBER AND INFORMATION ON THE APPLICATION OF ARTICLE 67(3) EPC

The Receiving Section hereby informs you that the technical preparations for publication of the above-mentioned European patent application have been completed.

The provisional protection under Art. 67(1) and (2) EPC in the individual Contracting States becomes effective only when the conditions referred to in Art. 67(3) EPC have been fulfilled (for further information, see EPO brochure "National Law relating to the EPC").

This application will be published on 02.01.04 without the European search report. The publication will be mentioned in European Patent Bulletin number 2004/01

The publication number is: 1376977

The title of the invention in the three official languages of the European Patent Office is worded as follows:

Verteiltes Rechnersystem für Vorrichtungsressourcen basierend auf Identität

Identity-based distributed computing for device resources

Système informatique répartie pour ressources de dispositif basé sur l'identité

In all future communications to the EPO, please quote the application number as indicated above, i.e. including the final four figures (which identify the Directorate responsible for the subsequent procedure). Amendments to a European patent application or European patent must be filed in the language of the proceedings.

REMARK: An issue of the published European patent application will be forwarded to you directly from our printer.

RECEIVING SECTION



3A6

673

GRÜNECKER KINKELDEY STOCKMAIR & SCHWANHÄUSSER

ANWALTSSOZIELTÄT

GKS & S. MAXIMILIANSTRASSE 56 D-80538 MÜNCHEN GERMANY

EUROPEAN PATENT OFFICE
ERHARDTSTR. 27

80298 MUENCHEN

RECHTSANWÄLTE LAWYERS

MÜNCHEN
DR. HELMUT EICHMANN
GERHARD BARTH
DR. ULRICH BLUMENRÖDER, LL.M.
CHRISTIAN NIKLAS-FALTER
DR. MAXIMILIAN KINKELDEY, LL.M.
DR. KARSTEN BRANDT
ANJA FRANKE, LL.M.
UTE STEPHANI
DR. BERND ALLEKOTTE, LL.M.
DR. ELVIRA FRANG, LL.M.
KARIN LOCHNER
BABETT EITLE
CHRISTINE NEUHIER
SABINE PRÜCKNER

PATENTANWÄLTE EUROPEAN PATENT ATTORNEYS

MÜNCHEN
DR. HERMANN KINKELDEY
PETER M. JAKOB
WOLFGANG MEISTER
HANS HILGERS
DR. HENNING MEYER-PLATH
ANNELE EINHOLD
THOMAS SCHUSTER
DR. KARA GOLDSACH
MARTIN AUFENANGER
GOTTFRED KLITZSCH
DR. HEIKE VOGELSANG-WENKE
REINHARD KNAUER
DIETMAR KUHL
DR. FRANZ-JOSEF ZIMMER
BETTINA K. REINHOLD
DR. ANTON K. BEAU
DR. UDO WEIGELT
RAINER BERTRAM
JENS KOCH, M.S. (U.G.P.A.) M.S.
BERND ROTHMEL
DR. DANIELA KINKELDEY
THOMAS W. LAUBENTHAL
DR. ANDREAS KAYSER
DR. JENS HAMMER
DR. THOMAS ECKELKAMP
JOCHEN KILCHERT
DR. THOMAS FRIEDL

PATENTANWÄLTE EUROPEAN PATENT ATTORNEYS

BERLIN
PROF. DR. MANFRED SCHULZ
DR. PATRICK ESKA M.S. LL.M.
KÖLN
DR. MARTIN DRUPMANN
CHEMNITZ
MANFRED SCHNEIDER

OF COUNSEL PATENTANWÄLTE

AUGUST GRÜNECKER
DR. GUNTHER BECKOLD
DR. WALTER LANCHOFF

DR. WILFRIED STOCKMAYER
(1992)

EPO - Munich
59

24. Juni 2003

IHR ZEICHEN / YOUR REF.

UNSER ZEICHEN / OUR REF.

DATUM / DATE

EP 27314/no

24.06.03

European Patent Application No.: 03 009 717.4
Applicant: MICROSOFT CORPORATION

The following documents are submitted herewith in response to the Office Action of
June 11, 2003

- Designation of Inventor
- Priority Document

(F.-J. Zimmer)

Encls.

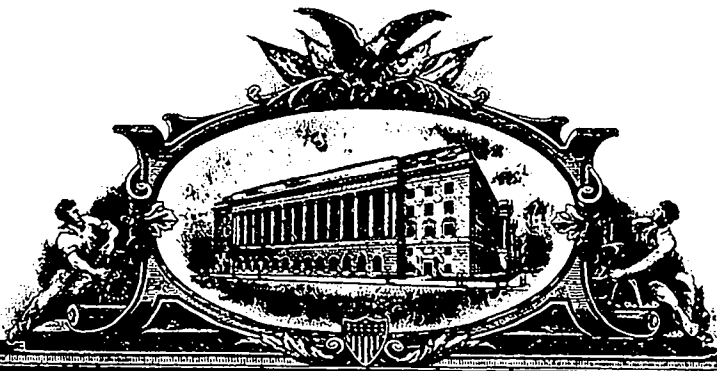
GRÜNECKER KINKELDEY
STOCKMAIR & SCHWANHÄUSSER
MAXIMILIANSTR. 58
D-80538 MÜNCHEN
GERMANY

TEL. +49 89 21 23 50
FAX +49 89 22 02 87
FAX +49 89 21 86 92 93
http://www.grunecker.de
e-mail: info@grunecker.de

DEUTSCHE BANK MÜNCHEN
No. 17 51734
BLZ 700 700 10
SWIFT: DEUTDE33

347

PA 1016898



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

May 30, 2003

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: 10/187,221

FILING DATE: June 28, 2002



By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

T. LAWRENCE
Certifying Officer

675

07-02-02 10187221

06/28/02
JCE12 U.S. PTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Attorney Docket No. 3020

U.S. PTO
10/18/22

TRANSMITTAL LETTER

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Transmitted herewith for filing by Express Mail is the patent application of David ALLES and George M. MOORE

Title: *Identity-Based Distributed Computing for Device Resources*

1. An application consisting of 50 pages of cover sheet, specification, claims and abstract.
2. Seven (7) sheets of formal drawings are attached
3. 3 pages executed Declaration of the Inventors
4. 1 page Recordation Form Cover Sheet ✓
5. 3 pages executed Assignment of the Inventors
7. The following additional documents are enclosed: Credit Card Payment Form, Certificate of Express Mail, Return Postcard.
8. Please publish this application in the name of:
MICROSOFT CORPORATION, REDMOND, WASHINGTON.
9. The fee has been computed as follows:

Filing Fee							\$740
Total Claims Filed	29	Minus Number Included	20	Equals Amount Extra	9	\$18	\$162
Indep. Claims Filed	4	Minus Number Included	3	Equals Amount Extra	1	\$84	\$84
TOTAL FEE							\$986



349

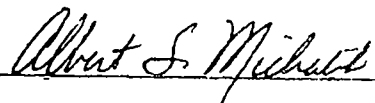
ALBERT S. MICHALIK

- ☒ Charge the Credit Card identified on the attached Credit Card Payment Form the sum of \$1,026.00; (\$986.00 as calculated above plus \$40.00 for the Assignment Recordation Fee).
- ☒ If any additional fee is required in connection with this response, please charge Deposit Account No. 50-1618.
- ☒ If any overpayment has been made in connection with this response, please credit Deposit Account No. 50-1618.

Please address all further correspondence to:

Law Offices of Albert S. Michalik, PLLC
704 -228th Avenue NE
Suite 193
Sammamish, WA 98074
Tel: (425) 836-3030
Fax: (425) 836-8957

Respectfully submitted,



Albert S. Michalik
Registration No. 37,395

3020 transmittal

JUL 18 2002 10:06 AM

In re Application of ALLES et al.
Attorney Docket No. 3020

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" mailing label number EU100107115US

Date of Deposit: June 28, 2002

I hereby certify that the following documents:

New Utility Application in the name of David ALLES and George M. MOORE for: "Identity-Based Distributed Computing for Device Resources," including:

1 page Cover Sheet, 40 Pages Specification, 8 Pages Claims, 1 Page Abstract, 7 Sheets Formal Drawings, 2 Pages Transmittal, 1 Page Credit Card Payment Form, 3 Pages Executed Declaration of Inventors, 1 Page Recordation Form Cover Sheet, 3 Pages Assignment of Inventors, and Return Postcard

are being deposited with the United States Postal Service "Express Mail Post Office To Addressee" Service under 37 C.F.R. 1.10 on the date indicated above and is addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231.

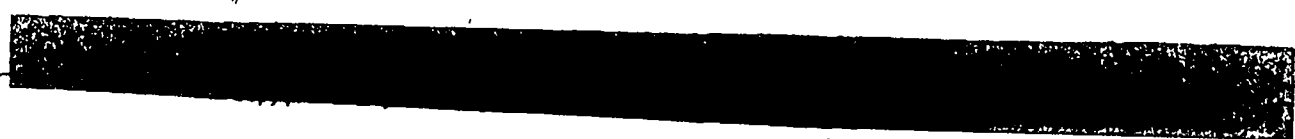
Albert S. Michalik

(Typed or printed name of person mailing paper or fee)

Albert S. Michalik

(Signature of person mailing paper or fee)

3020 express mail cert



10187221 .062802

S P E C I F I C A T I O N

TO ALL WHOM IT MAY CONCERN:

Be it known that we, David Alles, a citizen of Canada, residing at 4123 Evanston Avenue N. Seattle, Washington 98103 and George M. Moore, a citizen of the United States, residing at 5565 248th Place SE, Issaquah, Washington 98029 have invented a certain new and useful IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES of which the following is a specification.

IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

FIELD OF THE INVENTION

The invention relates generally to computer systems, and
5 more particularly to computer resources.

BACKGROUND OF THE INVENTION

In early personal computing systems, application program
had to be programmed with knowledge of various computing
10 devices in order for the application to use such devices. For
example, in order to print to a particular printer, a word
processing program had to know how to print to the printer,
and generally supplied a driver for each supported printer.
Alternatively, a printer would supply drivers with it for the
15 most popular programs. The user had to configure each program
to inform that program as to which printer was connected to
the personal computer.

Contemporary operating systems such as Microsoft Windows®
provided developers with a way to interface applications with
20 such devices regardless of the specific device driver needs
and requirements. For example, with the Microsoft Windows®
operating system a user or the device tells the operating
system which printer (or printers) is connected, after which
any application is able to communicate with the printer driver
25 via the operating system. As a result, to accomplish

printing, application programs generally only need to deal with a set of standardized printing functions provided by the operating system. The enormous benefits of such an abstraction cannot be understated, e.g., this abstraction enables any one of thousands of software programs to interact with any one of thousands of printers and/or other devices, without the program having to know (or otherwise supply) the details about the printer in advance.

Microsoft Corporation's .NET technology provides a somewhat similar type of abstraction regarding a user's or company's data. In general, with .NET technology, any .NET-capable program on any device can interact with a user's data via centralized schema-based services. Because of the schematization, the programs do not need specific knowledge about the details of that data, e.g., which used to be stored in proprietary file formats, but rather only needs to be able to deal with (read and write) data formatted according to one or more of the schemas.

With .NET technology, a user's (or other entity's) identity controls access to the centrally-maintained data, e.g., a user may use a .NET Inbox service keep his or her electronic mail data centrally stored (e.g., on the Internet or an intranet), and then access that data via the .NET inbox service, at any time, from any device capable of connecting to

10187221 .062002

the Internet or an intranet and having a program capable of interacting with the .NET Inbox service. The user's identity is used to securely match the user with the appropriate data. Users can also controllably allow access to data by other
5 users, e.g., a user can selectively make certain business contact information available to business associates, make personal contact information available to friends, and so on, based on the others' identities.

While .NET thus provides tremendous data-based benefits
10 to users, companies and other entities, the centralized service and centralized data are not particularly concerned with how a user uses various devices and those devices' resources. For example, devices such as a personal computer and pocket personal computer have, for quite some time, been
15 able to directly synchronize with one another. To an extent, .NET technology is another way to keep such data synchronized, although indirectly, without needing specialized and proprietary synchronization programs.

However, even with .NET, a user cannot easily use one
20 device to access the resources of another device, such as to use the computing power, storage, bandwidth, input and output mechanisms, programs and/or other resources of one device on behalf of another device, e.g., to perform a complex or better-suited task on the more powerful device. What is

101137221 .062802

needed is a common, consistent way to remotely discover and expose computing resources of various computing devices to one another for cross-resource usage.

5

SUMMARY OF THE INVENTION

Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.

15 Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus abstracted into its resources, which are then programmatically

addressable from other devices in a composable fashion (e.g., the remote resources may be defined in such a way that they are, in turn, comprised of smaller resources contained within the larger definition of the resource) . In other words, for
5 devices to participate in an intelligent distributed computing model, various aspects of each participating device are abstracted and made programmatically addressable from other devices. A unified identity-based mechanism discovers the list of remote resources on the device. A set of one or more
10 schemas consistently describes the remote resources on the device, and a unified security model allows the owner of the device to control who has access to which resources, and to what extent.

With the present invention, desirable computing solutions
15 for a great number of example scenarios are enabled, including a set of real-time scenarios in which one user device interacts with one or more remote computing devices in real-time, and a set of scheduled or predicted scenarios that operate transparently and automatically on a user's behalf.

20 Other advantages will become apparent from the following detailed description when taken in conjunction with the drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1A is a block diagram generally representing a computer system into which the present invention may be incorporated;

5 FIG. 1B is a block diagram generally representing another computer system into which the present invention may be incorporated;

FIG. 2 is a block diagram representing various components in a distributed computing environment in accordance with an
10 aspect of the present invention;

FIG. 3 is a block diagram generally representing an architecture for discovering and exchanging resource information in accordance with an aspect of the present invention;

15 FIGS. 4 and 5 are block diagrams representing examples of how the present invention can improve a user computing experience by distributing computing operations to resources, in accordance with an aspect of the present invention;

FIG. 6 is a flow diagram representing exemplary logic and
20 related operations performed to facilitate the sharing of resources in a distributed computing environment in accordance with an aspect of the present invention; and

FIG. 7 is a block diagram generally representing an example implementation including resources distributed in accordance with an aspect of the present invention.

5

DETAILED DESCRIPTION

EXEMPLARY OPERATING ENVIRONMENTS

FIGURE 1A illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one
10 example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components
15 illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for
20 use with the invention include, but are not limited to: personal computers, server computers, hand-held or laptop devices, tablet devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe

10197221 .062302

computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth, that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in local and/or remote computer storage media including memory storage devices.

With reference to FIG. 1A, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and

not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus. Note that any and all of the elements shown in computer 110 may be connected to each other by standard high-speed computer networks, including those distributed across a wide geographic area. For example, the system memory 130 may be located in a physically different location than processing unit 120, but yet defined in a composable fashion as logical computer 110.

The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 110 and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,

digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can

5 accessed by the computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data

10 signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media

15 such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read

20 only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or

program modules that are immediately accessible to and/or
presently being operated on by processing unit 120. By way of
example, and not limitation, FIG. 1A illustrates operating
system 134, application programs 135, other program modules
5 136 and program data 137.

The computer 110 may also include other removable/non-
removable, volatile/nonvolatile computer storage media. By
way of example only, FIG. 1A illustrates a hard disk drive 141
that reads from or writes to non-removable, nonvolatile
10 magnetic media, a magnetic disk drive 151 that reads from or
writes to a removable, nonvolatile magnetic disk 152, and an
optical disk drive 155 that reads from or writes to a
removable, nonvolatile optical disk 156 such as a CD ROM or
other optical media. Other removable/non-removable,
15 volatile/nonvolatile computer storage media that can be used
in the exemplary operating environment include, but are not
limited to, magnetic tape cassettes, flash memory cards,
digital versatile disks, digital video tape, solid state RAM,
solid state ROM, and the like. The hard disk drive 141 is
20 typically connected to the system bus 121 through a non-
removable memory interface such as interface 140, and magnetic
disk drive 151 and optical disk drive 155 are typically
connected to the system bus 121 by a removable memory
interface, such as interface 150.

The drives and their associated computer storage media, discussed above and illustrated in FIG. 1A, provide storage of computer-readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1A, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as mouse, trackball or touch pad. Other input devices not shown in FIG. 1A may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected

691

10107221 062802

to the system bus 121 via an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 195 and printer 196, which may be connected through an output peripheral interface 194 or the like.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1A. The logical connections depicted in FIG. 1A include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. For example, in the present invention, the computer system 110 may comprise source machine from which data is being migrated,

365

and the remote computer 180 may comprise the destination machine. Note however that source and destination machines need not be connected by a network or any other means, but instead, data may be migrated via any media capable of being
5 written by the source platform and read by the destination platform or platforms.

When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the
10 computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160 or other appropriate mechanism. In a networked
15 environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1A illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network
20 connections shown are exemplary and other means of establishing a communications link between the computers may be used.

FIGURE 1B illustrates an example of another suitable operating environment (computer device 188) in which at least



part of the present invention may be implemented. As with
FIG. 1A, the operating environment of FIG. 1B is only one
other example of a suitable operating environment and is not
intended to suggest any limitation as to the scope of use or
5 functionality of the invention. The other computer device 188
in FIG. 1B typically includes at least some form of computer
readable media. Computer-readable media can be any available
media (as generally defined above) that can be accessed by the
computing device 188.

10 FIG. 1B shows functional components of a computing device
188 such as a handheld (pocket-sized or tablet) personal
digital assistant, including a processor 189, a memory 190, a
display 192, and a keyboard 193 (which may be a physical or
virtual keyboard). The memory 190 generally includes both
15 volatile memory (e.g., RAM) and non-volatile memory (e.g.,
ROM, PCMCIA cards, and so forth). An operating system 194 is
resident in the memory 190 and executes on the processor 189,
such as the Windows® CE operating system from Microsoft®
Corporation, or another operating system.

20 One or more application programs 195 are loaded into
memory 190 and run on the operating system 194. Examples of
applications include email programs, scheduling programs, PIM
(personal information management) programs, word processing
programs, spreadsheet programs, Internet browser programs, and

so forth. The handheld personal computer 188 may also include a notification manager 196 loaded in the memory 190, which executes on the processor 189. The notification manager 196 handles notification requests, e.g., from the application
5 programs 195.

The handheld personal computer 188 has a power supply 197, which is implemented as one or more batteries. The power supply 197 may further include an external power source that overrides or recharges the built-in batteries, such as an AC
10 adapter or a powered docking cradle.

The exemplary handheld personal computer 188 represented in FIG. 1B is shown with three types of external notification mechanisms: one or more light emitting diodes (LEDs) 198 and an audio generator 199. These devices may be directly coupled
15 to the power supply 197 so that when activated, they remain on for a duration dictated by a notification mechanism even though the handheld personal computer processor 189 and other components might shut down to conserve battery power. The LED 198 preferably remains on indefinitely until the user takes
20 action. Note that contemporary versions of the audio generator 199 use too much power for today's handheld personal computer batteries, and so it is configured to turn off when the rest of the system does or at some finite duration after activation.

IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

The present invention is generally directed to a system and method that facilitate the distribution of resources of devices to other devices. The present invention is ordinarily not involved with what the devices do with each other's resources, what is communicated at that time, or how the communication between devices takes place. Rather, the present invention can be generally considered more as a structured way for devices to find out about each other's resources, capabilities and other details, and then allow the devices to share resources on their own. Although not necessary to the present invention, as part of facilitating the resource distribution, security is provided for practical purposes, at least to ensure that valid credentials are required to use another device's resources. Thus, the present invention is preferably identity-based, in that distributed resources have relationships with one another according to identity data (e.g., credentials) provided to the devices.

In general, the present invention is directed to distributing computing among a user's devices, wherein a user (or owner / administrator) is anyone authorized to control and/or use the devices. However, because in general devices are increasingly becoming disaggregated, the methods and

systems described herein may be more properly described as distributed computing of resources. For example, a single distributed device may be made up of many resources, but may share as little as one its resources (e.g., only its non-

5 volatile storage) to participate as a resource provide. Notwithstanding, as used herein, the terms "device" and "resource" are generally interchangeable except where otherwise apparent, e.g., using another device or using a resource of the other device to accomplish some computing task

10 are generally considered the same thing. Also, as used herein, bandwidth is a device resource, even though bandwidth may not ordinarily be considered as being associated with any particular device.

As generally represented in FIG. 2, a distributed

15 computing environment 200 constructed in accordance with various aspects of the present invention includes a first device 202 (Device A, such as the computer system 110 of FIG. 1A) and a second device 212 (Device B, such as the computer system 188 of FIG. 1B), along with possibly other devices (not

20 shown). Each device that participates in the distributed computing environment 200 includes a distributed computing manager, 204 and 214. In general, and as described below, the distributed computing managers 204 and 214 may comprise some amount of code that communicate with one another over some



medium, e.g., wireless FM, line-of-sight such as infrared,
satellite, a direct connection, the Internet, or an intranet.
Indeed, more than one such medium may be available, and the
distributed computing managers 204 and 214 may choose which
5 one or ones are best for a given circumstance.

An initial purpose of the distributed computing managers
204 and 214 are to locate each other. Such location may be
automatic, such as by having each device occasionally
broadcast its availability to others and/or by polling for the
10 availability of other devices, or alternatively may operate in
response to some specific request from a program (or user) to
connect to another device. As represented in FIG. 2, the
distributed computing managers 204 and 214 each include a
security mechanism 206 and 216, respectively, that exchange
15 credentials 207, 217 or the like (e.g., obtained from an
authentication service 220) to ensure that the user who is
currently operating the device which is contacting it (and/or
the device itself), is valid in the distributed computing
environment, and similarly, if necessary, that the device
20 which is being contacted is valid. Note that a program being
used may also factor into the security / access control, e.g.,
a first program running on one device may have authority to
use another device's resource or resources, but not another
program, independent of or in addition to the user identity

698

101137221 .062802

and/or the device identity. Further note that the security mechanism may operate per identity and per resource, e.g., one entity may be authorized to use a device's display, but not its storage, while another entity may be able to use any of
5 that same device's available resources, including its storage. This is similar to the .NET security model in which services provide methods that control access to a users' data based on the requesting user's identity, defined role and scope for that role. In this way, data can be accessed by its owner,
10 and shared to an extent determined by the owner. In the present invention, instead of data it is a device resource being accessed, whereby a user's identification, defined role and scope for that role determine the extent to which a resource may be accessed.

15 Indeed, the security mechanisms of the present invention are based on those essentially those employed for security / access control in .NET services, as generally described in United States patent applications, serial nos. 10/003,754 entitled "IDENTITY-CENTRIC DATA ACCESS" and serial number
20 10/003,767, entitled "AUTHORIZING A REQUESTING ENTITY TO OPERATE UPON DATA STRUCTURES" filed on October 22, 2001, assigned to the assignee of the present application, and herein incorporated by reference. To this end, at least a thin subset of the .NET service runs on each device which

372

expects to advertise and share computing resources in the environment 200. In addition, a localized internal consistent security model may be executed on these devices such that local security is provided even when two local peer-to-peer
5 devices may be disconnected from the centralized (e.g., Passport) authentication service 220 (FIG. 2).

The present invention is not limited to any particular transmission medium, but provides that the distributed resource services may be directly addressable to the target
10 device, or may be queried via the Internet or an intranet, such as via the centralized set of .NET services on the web. When connected via the Internet or intranet, a program may query the last known computing resources available on any device associated with the user's identity, even when the
15 device is offline or otherwise inaccessible. Bandwidth is also a resource, and may impact how a device's resources are shared, e.g., it may be more efficient to leverage a personal computer's resources when the personal computer has a high-speed connection to the Internet, but not if the personal
20 computer connects only at low speed, such as if the other device has a high-speed connection. In fact, the personal computer may want to leverage the high-speed connection of the other device, in which event that other device may then leverage a resource of the personal computer, e.g., the

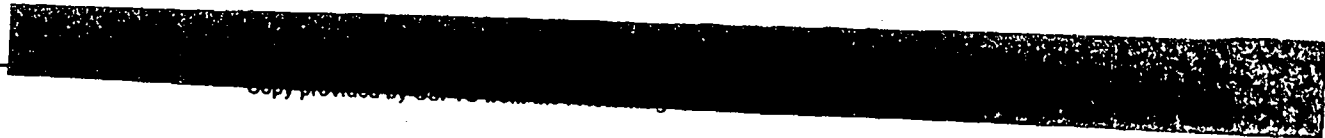
700

10.1 BY 221 .06.2002

devices exchange resources to provide a more-optimal user experience.

In accordance with one aspect of the present invention, the distributed computing managers 204 and 214 also each
5 include a respective discovery mechanism, 208 and 218, that allows the devices to discover another participating device's shared resources / capabilities, that is, when that other device is configured to share its resources. The respective resource data 209 and 219 is formatted (e.g., in XML) and
10 exchanged in accordance with a defined schema, as described below. As represented in FIG. 2 by the exchange of the schematized data 209 and 219, both devices 202 and 204 are configured to share their resources ($R_{A1}-R_{Am}$ and $R_{B1}-R_{Bn}$) with other devices in the distributed computing environment.
15 However as can be appreciated, in a given environment, a particular device (e.g., a mobile telephone) may be only a consumer of resources rather a provider of any resource, or conversely may be only a provider, and not a consumer.

A device that is sharing its resources need not directly
20 communicate the resource data with the other device, but instead, for example, can advertise its resources somewhere for other devices to discover. For example, a .NET service (.NET Devices) is provided that centrally stores device resource capabilities such that data may be appropriately



374

formatted before sending it to a device, e.g., so as to not
 send high-resolution graphics to a device incapable of
 displaying them. A similar .NET Resources service, or an
 extended .NET Devices service, may be provided to advertise
 5 this information to other devices.

The following tables set forth some example schemas that
 may be used (e.g., with XML-formatted data) to exchange
 resource (storage and processing power) information regarding
 a user's devices, which in this example include work and home
 10 personal computers, and a cellular telephone:

```
<myStorage>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
</myStorage>
```

```

<myProcessingPower>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
</myProcessingPower>

```

As is understood, the schema is preferably resource based, i.e., multiple devices may be listed for each resource type, to match the general concept of an identity having distributed resources available for use, independent of the actual device on which the resource exists. Notwithstanding, the present invention is not limited to any particular schema and/or formats, e.g., the schemas may be device-based, such that each device may instead provide its list of available resources to other devices such that the same information is known.

As generally represented in FIG. 3, an architecture 300 is provided in order for devices to participate in an intelligent distributed computing model, in which various

703

10137221.0162302

aspects of the participating devices are abstracted and made
programmatically addressable from a different device. A
universal plug-and-play (UPnP 2.0) infrastructure or the like
may be used as the basic, low-level mechanism 302 for
5 discovery of these resources. Alternatively, the list of
resources for various computing devices may be cached in some
well-known location, such as an Active Directory store, to
enable querying of remote resources even if the device is only
accessible through some intermediary such as a firewall or
10 wireless hub. Atop the discovery mechanism are the schemas
304, e.g., one schema associated with each specific type of
the computing resource, with each of those schemas exposed,
for example, as in the .NET architecture.

Note that while UPnP provides a suitable discovery
15 mechanism 304, it is only part of the process of actually
using the remote computing resources. Thus, UPnP and .NET
technologies are used to facilitate resource sharing, but are
not used to actually employ the resources available on a
device. Rather, once the resources are known, appropriately
20 focused protocols for that specific resource domain are used.
By way of example, UPnP, with the addition of the .NET
identityHeader, endpointHeader, and requestHeader SOAP
semantics, is well-suited to obtain the video screen
characteristics of the remote device, however a protocol such

397

as RDP will be used to transfer video data after the initial discovery and handshaking occurs between the devices. Note that this is similar to the way .NET approaches data services today, e.g., .NET provides a mechanism to connect to an

5 electronic endpoint representing a proprietary (e.g., Messenger) protocol, but once that endpoint is connected, .NET allows the client to use an existing real-time protocol. Other protocols appropriate for other types of resources are used, e.g., CIFS+ for storage, which scales well to both

10 corporate and home storage scenarios and enables the user's file system to continuously expand to other storage devices, without forcing the user to deal with drive letters, namespaces, and so forth. In summary, the present invention provides an identity-based approach to discovering these

15 additional resources that are available for use, but does not thereafter directly deal with their use.

FIG. 6 is an example flowchart that represents some general aspects of the present invention, beginning with step 600 which represents receiving a request to use a shared

20 resource. The request may be general, e.g., requesting the list of resources that a user has available on devices configured to participate, or may be a request for a specific resource, e.g., in response to an advertisement. For purposes

705

11137221.062002

of this example, the request is treated as a request for one specific resource.

Step 602 represents the evaluation of the credentials, such as provided by the Microsoft .NET Passport authentication service. The request may include the credentials, or the credentials may be separately exchanged, e.g., before, as part of, or after step 600. If not valid at step 602, step 602 branches to step 604 which represents rejecting the request. Note that depending on a given environment, the request without proper credentials may be ignored rather than specifically rejected.

If valid, step 606 is executed, and represents determining the amount of the requested resource that is available. For example, a resource such as storage may be limited because it has only limited free space, and further, a device may be configured to make only part of its storage space available for sharing, reserving the rest for itself. A resource such as processor cycles may be made available only to the extent the processor is not being otherwise used, and even then may be further limited so that another device can only take up to a maximum amount of the unused cycles. Any resource may have a time-out value associated with it, which may be further contingent upon its use, e.g., storage can be shared for no more than one hour per request, and the request

379

706

10187221 .061202

will be terminated if no file system requests are detected at least every two minutes. In this manner, a device may be configured to share its resources to an extent controlled by its owner / administrator.

5 Note that the resource access may be dynamic, e.g., unused processor cycles may be shared, however CPU usage is a number that fluctuates, and the requesting guest device may be prioritized along with other devices and/or processes on the device that is sharing the CPU. For example, guest users may
10 only consume a controllable percentage of resources when the owner of the device is not otherwise using them, such as bandwidth or processing power. Both CPU and bandwidth are examples of resources which are scheduled according to time
15 time-based resources such as disk or memory usage. As a result, the scheduling mechanisms for time-based resources in general will be fundamentally different than for non-time-based resources. Note that kiosk locations would enable many scenarios that would be useful to such guests. A resource
20 such as a display monitor may be either entirely available for sharing or not at all, depending on whether or not it is presently being used. Thus, although not represented in FIG. 6, step 606 may also reject (or ignore, if appropriate) a request when there is no amount of a requested resource

380

707

10187221 .062807

available for sharing. Step 608 represents returning a response to the requesting entity, preferably the schema-based response that provides the resource capability data to the requesting entity.

5 Step 610 represents allowing the requested resource to be shared, and includes, for example, the exchange of data between the requesting device and the sharing device so that the sharing performs some corresponding task for the requesting device. Step 612 tests whether the resource was
10 exhausted, e.g., timed out or is no longer available. Note that this is shown as a loop in FIG. 6 for purposes of simplicity, however it is readily understood that the actions may be event driven. Further, a step (such as step 606) may be included in the loop / used to fire the event, e.g., the
15 resource usage is dynamically monitored to determine when it is exhausted for the particular request. Note that a resource may be directly exhausted when the requesting entity informs the device that the resource is no longer needed.

Step 614 represents notifying the requesting device that
20 the resource is no longer available for its use. Step 616 represents disabling the resource usage, so that, for example, once exhausted the requesting device will need to make a new request if it wants to share resources.

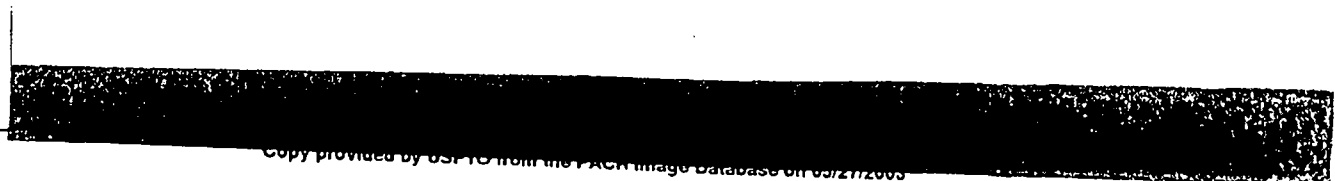
381

708

FIGS. 4 and 5 provide examples of some of the many possible uses of the present invention. As will be understood from the examples of FIGS. 4 and 5, integrating computational proxy support into a network edge device or appliance may
5 result in a better user experience for users of an internal network.

In a first example represented in FIG. 4, a small mobile device 400 (such as a mobile telephone) wants to print to a printer 402, however the small device 400 does not have an
10 appropriate printer driver. This may be because the mobile device 400 does not have sufficient memory to install and maintain a printer driver needed for that printer 402, and/or does not have an easy way for the needed printer driver to be installed, and/or for other reasons. Note that this is a real
15 scenario, and one proposed solution is to emit XHTML-PRINT as a simplified PDL (Page Description Language) from the mobile device 400 to the printer 402, e.g., over some network connection (shown via the dashed line in FIG. 4). Although XHTML-PRINT is adequate to an extent, this solution has a
20 number of drawbacks, e.g., XHTML-PRINT is not WYSIWYG, cannot control line breaks, and generally provides a poor user experience.

However, if via the present invention, the mobile device 400 first dynamically discovers a personal computer 404



782

409

10187221 .062002

equipped with a proper printer driver 406, (and which the mobile device user is authorized to use), the mobile device 400 instead provides printing data via the personal computer 404 to the printer driver 406, which in turn will create a

5 WYSIWYG output using PCL, Postscript or rasterization. As mentioned above, the present invention provides the discovery mechanism, and thereafter the mobile device 400 may communicate in any way with the personal computer 404, such as by a lightweight PDL that has much higher-fidelity (relative

10 to XHTML-PRINT). The result is high-quality output rendered very quickly, by leveraging the faster processor and proper printer driver 406 in the personal computer 404.

FIG. 5 demonstrates a second example, generally directed to providing an improved user experience when operating a PDA

15 (personal digital assistant) 500 that is to be executing managed code remotely on a server 502 (e.g., an ASP.NET server). One way to accomplish this is to have the PDA connect to the server to run the managed code, (e.g., by leveraging Mobile Controls support in ASP.NET), with the

20 results transmitted as HTML, as indicated by the dashed line in FIG. 5. The server 502 dynamically scales the HTML content to the appropriate screen size of the device 500, without the ISV creating special-purpose code for each device 500. However, this requires many round trips with the server 500,

383

710

10187771 .062312

with the screen flashing for updates, and generally provides a degraded user experience.

An improved solution provided by the present invention is accomplished by having the PDA dynamically discover that a
5 personal computer 504 associated with the user's identity can be a proxy for a number of computational resources, as generally represented in FIG. 5. In such an event, the PDA issues an HTML request for a remote page via the proxy personal computer 504. The personal computer 504 intercepts
10 the request and starts a session with the remote server 502 to offload the UI rendering managed code, e.g., using WSDL as the protocol to the server 502. Typically, the bandwidth from the proxy personal computer 504 to the server 502 will be less than the bandwidth from the PDA device 500 to the proxy
15 personal computer 504, whereby the smaller amount of data transferred over the slower link to and from the server 502 results in a more responsive user interface when displayed on the PDA 500, providing an improved user experience. Of course, the two bandwidths can be first evaluated to ensure
20 that the proxy solution is the more-optimal one.

Further, note that if the PDA 500 was equipped with locally-running controls that the server was able to leverage, an even better experience may be provided. For example, ASP.NET automatically leverages any local controls (e.g., such

384

711

10187221 .062102

as a date picker control) running locally on a PDA running a Windows®-based operating system. This results in even less round trips to the server 502 and/or proxy 504, with less resultant screen updating / flashing.

5 As can be appreciated, there are numerous other examples that are enabled by having computing resources revolve around an identity, instead of around the particular devices those resources ship with. The resource schemas and service fabric are not limited to simply describing the resources, but also
10 include elements to assist those resources working together. This can be appreciated from the following two types of scenarios, namely a set of real-time scenarios in which the user is interacting with a remote computing device in real-time, and a set of scheduled or predicted scenarios. An
15 example of a predicted scenario may include a storage load balancing mechanism that occasionally runs on a home network, to make sure that one computing device's free storage space is not going to run out while another drive has plenty of room.

 Thus, a rich schema may provide data that goes beyond
20 describing the resources themselves, such as methods to schedule, at a particular time or at regular intervals, prioritization machinery, and data indicating how to deal with possible contention between real-time and predicted tasks, or multiple such tasks. The schema may provide for the concept

385

- of cost, e.g., if two connections are available to the Internet, it is valuable to recognize that the user may be paying one price for unlimited access but another price for an amount of data transferred over a different mechanism, and
- 5 select the better option.

- Replication/sharing by a logging system may be used to understand how users are interacting with their devices, and to achieve a consolidated view of how a user operates, rather than have only disparate views separated by device. This data
- 10 is appropriately known to many devices, and will provide an automatic way for such data to be consolidated. Prediction optimizations may be saved and exchanged based on monitored usage, e.g., if the system knows that a user manually
- 15 downloads particular information to a hand-held computer every Friday before leaving work, the system can start the download during device idle time so the "synchronization" appears instantaneous. End user options/settings/preferences is also valuable information for the shared environment, e.g., what is a desirable or even necessary feature for some users is
- 20 undesirable or even annoying to others. By maintaining this data, the end user has the power to be in control of their experience. Also, in a typical environment, there will be at least one rich client that uses the protocol/schema to perform system management, where "system" refers to the overall set of

resources associated with an identity. Algorithms and methods can use the schema for predicting such operations.

As is understood, there are many valuable scenarios enabled by through abstracting device resources and aggregating them around an identity. A number of examples are provided, however the list should not be considered exhaustive or in any way limiting, but rather serves to illustrate some of the many benefits and advantages of the present invention. As examples mentioned above, a device with a small computer screen may leverage an unused computer monitor connected to a personal computer for better display, use resources of a proxy to facilitate a better display on its own display (FIG. 5) or print to a printer even without printer driver, by connecting to a printer via a personal computer that does have the correct driver (FIG. 4).

Other examples can be seen in a very low-powered device, such as a mobile telephone, performing complex computations by using the memory and CPU of a personal computer. Stored data can be automatically and transparently moved from a first hard drive to a second when the first hard drive is getting full, including, for example, from personal digital video recorders that record video programming and need to delete old shows to make room for more recent recordings. The present invention instead moves those older shows to another storage, rather

7d4

10187221 .062802

than delete them. A portable audio player need not be able to play any type of physical media, but can instead receive and buffer audio streamed to it from a personal computer and play the audio from memory.

5 Other examples include allowing users to easily access their files from any device regardless of where it is stored; for example, while on the road with only a small personal computer, being able to easily open files stored on a home personal computer or work personal computer, (independent of
10 centralized .NET data storage). A user may see all of the free space in various devices, including in various public sharepoints on a corporate network. A large-screen TV connected to an appropriate device may be used to view photographs saved on a personal computer folder, run a
15 slideshow, watch home movies or movie trailers saved on a personal computer, and so on.

FIG. 7 provides a demonstration of an implementation 700 run using four interconnected computing systems, 702-705. The implementation 700 includes a command shell 710 and
20 application program 712 (or similar function code) on the computing system 704, a keyboard resource 714 on the computing system 702, a storage (e.g., disk) 716 on the computing system 703, and a display 718 on the computing system 705. Thus, in this example the resources are distributed such that there is

388

715

10187221 OFFICE

an input resource the on computer 702, storage resource on the computer 703, and an output resource on the computer 704. The computer 704 acts as a coordinating authority for the implementation 700. Although the resources are on different
5 computer systems, the resources are accessed in accordance with the present invention to essentially form a single functioning system.

The shell resource 710 calls the application 712, and also connects to the other devices' resources through ports
10 724, 726 and 728, conceptually represented in FIG. 7 as one port per resource, although a separate input and output port may be provided. Ports essentially comprise structured queues that can be written (output) to or read (input) from via a handle to that port. Ports are mobile, data is not lost, and
15 security can be associated with a port such that no one except the intended entity will see the data intended for it.

The shell 710 used in the implementation 700 is based on a programming language referred to as xSpresso, as described in United States provisional patent application serial number
20 60/379,864, filed on May 10, 2002 and herein incorporated by reference in its entirety. In general, xSpresso is directed to a programming language and runtime that brokers control and data flows among asynchronous system components. Among other things, xSpresso expresses the cooperative dimensions of

389

716

10187221.062012

programs or services, including ordering and timing, that can be programmatically verified. The language may programmatically express the passing of an organizational scheme formed from a customizable tag-based language such as

5 XML.

In the example of FIG. 7, the numbered arrows labeled 1-4 and 6-10 represent an example of the various communications between the ports 724, 726 and 728 and the shell 710, to display the text data of a file in the storage 716 to the

10 display 718, e.g., similar to the well-known "type" function. In the example of FIG. 7, a user of the keyboard resource 714 initiates the operation by typing the word "type" on the keyboard 714 (arrow labeled one (1)). The shell 710 receives this via the port 724 and communicates data to the port 728

15 for display on the screen 718, (arrow labeled two (2)), e.g., to echo what the user has input and/or to prompt for a filename of a file that stores the text that the user wants to have displayed. In FIG. 7, the arrow labeled three (3) represents the user typing in the filename, such as

20 corresponding to an XML file, which may also be echoed (not separately labeled).

With the filename, the shell 710 then writes data (represented by arrow labeled four (4)) to the port 726 that corresponds to the storage (e.g., disk) resource 716, which

390

results in a read request to the storage's file system. In parallel, the shell 710 calls (the wide arrow labeled five (5)) the application program 712 to receive and output the read data. In response, the storage resource 716 begins
5 outputting data via port 726 to the application program 712, which in turn outputs it to the port 728 of the display resource 718, as represented by the arrows labeled six (6) and seven (7). The arrows labeled eight (8) and nine (9) represent the rest of the data being transferred in a similar
10 manner. The arrow labeled ten (10) represents the status (e.g., success) code being returned to the shell 710.

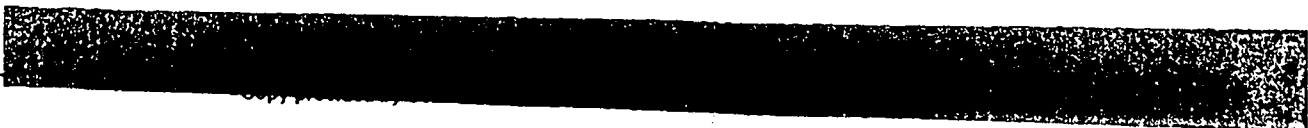
In keeping with the present invention, storage, input and output can be placed on different systems, with no change to the implementation. Legacy interoperability may be provided
15 via common language runtime (CLR) classes that represent common legacy components (e.g., the keyboard, console and storage). System dispersion is provided in a local and transparently distributed implementation, as is explicit handling of concurrency and message-passing.

20 As can be seen from the foregoing detailed description, there is provided a method and system for distributed computing of device resources based around the concept of identity. The method and system abstracts the resources of a computer from the individual devices in a manner that provides



numerous benefits to a user and generally improves the user's computing experience. The method and system are schema-based, and are extensible and highly flexible.

While the invention is susceptible to various
5 modifications and alternative constructions, certain
illustrated embodiments thereof are shown in the drawings and
have been described above in detail. It should be understood,
however, that there is no intention to limit the invention to
the specific forms disclosed, but on the contrary, the
10 intention is to cover all modifications, alternative
constructions, and equivalents falling within the spirit and
scope of the invention.



WHAT IS CLAIMED IS:

1. In a computing environment, a method comprising:
at a first computing device, discovering a second
computing device;
5 at the first device, obtaining data related to the second
device, the data including resource information of the second
device formatted according to a schema understood by the first
device; and
the first device interpreting the data to access a
10 resource of the second device.
2. The method of claim 1 further comprising,
communicating a request to use the resource from the first
device to the second device.
- 15 3. The method of claim 1 further comprising,
communicating identity information from the first device to
the second device.
- 20 4. The method of claim 3 wherein the identity
information corresponds to credentials.
5. The method of claim 4 wherein the credentials are
provided by an authentication service.

5 at a first computing device:

10 selecting a resource of a second device from the
 information; and

7. In a computing environment, a system comprising:

15 a first computing device associated with at least one identity, the first device including a first resource and configured to provide access to the first resource by at least one other device associated with the identity, the first computing device providing information about the first

20 resource according to a resource schema associated with the at least one identity;

- 42 -

6



6

• • • •

15

12

20

13. The system of claim 7 further comprising a third computing device connected to the first computing device, wherein the second device accesses the first resource to perform an operation with the third computing device.

722

10187221 .062802

14. The system of claim 7 wherein the third computing device comprises a printer, and wherein the first resource corresponds to a printer driver on the first computing device.

5

15. The system of claim 7 wherein the third computing device comprises a server, and wherein the first computing device acts as a proxy to the server for the second computing device.

10

16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.

15

17. A computer-readable medium having stored thereon a data structure, the data structure comprising:

a first field for identifying a type of resource;

a second field for identifying a device that includes a resource corresponding to the type of resource identified in the first field;

a set of at least one third field, each third field in the set corresponding to the second field for identifying

396

723

10187221 .062902

information about the device resource identified by the first field; and

wherein the data structure is used to provide resource information to another device that is different from the one
5 identified in the second field such that the other device can use a resource identified in the third field set.

18. The data structure of claim 17 wherein the first field includes information corresponding to a storage resource
10 type.

19. The data structure of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.

15

20. The data structure of claim 18 wherein the third field set includes information corresponding to free space data of the storage resource.

20 21. The data structure of claim 17 wherein the first field includes information corresponding to bandwidth data.

22. The data structure of claim 17 wherein the wherein the first field corresponds to processing power.

397

724

10187221.062002

23. The data structure of claim 22 wherein the third field set includes information corresponding to speed data of a processor.

5

24. The data structure of claim 22 wherein the third field set includes information corresponding to usage data of a processor.

10

25. The data structure of claim 22 wherein the third field set includes information corresponding to size data of a memory.

26. The data structure of claim 25 wherein the third field set includes information corresponding to usage data of a memory.

27. The data structure of claim 17 wherein the second field includes information identifying the device as a personal computer.

28. The data structure of claim 17 wherein the second field includes information identifying the device as a disk drive.

398

925

10187221 .062302

29. The data structure of claim 17 wherein the second field includes information identifying the device as a telephone.

5

389

726

10187221 .062802

ABSTRACT

A system, method and related data structures for discovering and describing computing resources available at various computing devices, and for exposing those resources as services that are addressable by software applications. The data describing the resources is arranged according to an identity-based schema. The computing resources may include, for example, storage capacity, bandwidth, processing power, input methods and mechanism, and rendering methods. The method and system are identity-based, whereby a user (with an identity) has access to the distributed resources commensurate with that identity.

200

727

PATENT APPLICATION

ATTORNEY DOCKET NO. 3020

MS DOCKET NO. 167497.1

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

the specification of which is filed herewith unless the following box is checked:

() was filed on _____ at US Application Serial No. or PCT International Application Number _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claims of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES NO
			YES NO

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE

U.S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS(patented/pending/abandoned)

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) listed below to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Add Attorney Name and Reg. No.	Albert S. Michalik, Reg. No. 37395	Daniel D. Crouse, Reg. No. 32022
	Kath E. Sako, Reg. No. 32628	

Send Correspondence to:

Contact Name: Albert S. Michalik
Firm Name: Law Offices of Albert S. Michalik, PLLC
Firm Address: 704 - 228th Avenue NE, Suite 393
City, State, and Zip: Sammamish, WA 98074

Direct Telephone Calls To:

Contact Name: Albert S. Michalik
Contact Phone Number: 425-836-9090

Aol

DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. 8120

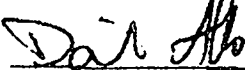
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: David Allen

Citizenship: Canada

Residence: 4123 Evanston Avenue N., Seattle, Washington 98103

Post Office Address: Same



Inventor's Signature

June 28, 2002

Date

Full Name of Inventor: George M. Moore

Citizenship: US

Residence: 5565 248th Place SE, Issaquah, Washington 98029

Post Office Address: Same

Inventor's Signature

Date

729

**DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION (continued)**

ATTORNEY DOCKET NO. 3020

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: David Alles

Citizenship: Canada

Residence: 4123 Evanston Avenue N., Seattle, Washington 98103

Post Office Address: Same

Inventor's Signature

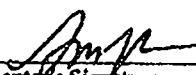
Date

Full Name of Inventor: George M. Moore

Citizenship: US

Residence: 5565 248th Place SE, Issaquah, Washington 98029

Post Office Address: Same



Inventor's Signature

6-28-02

Date

403

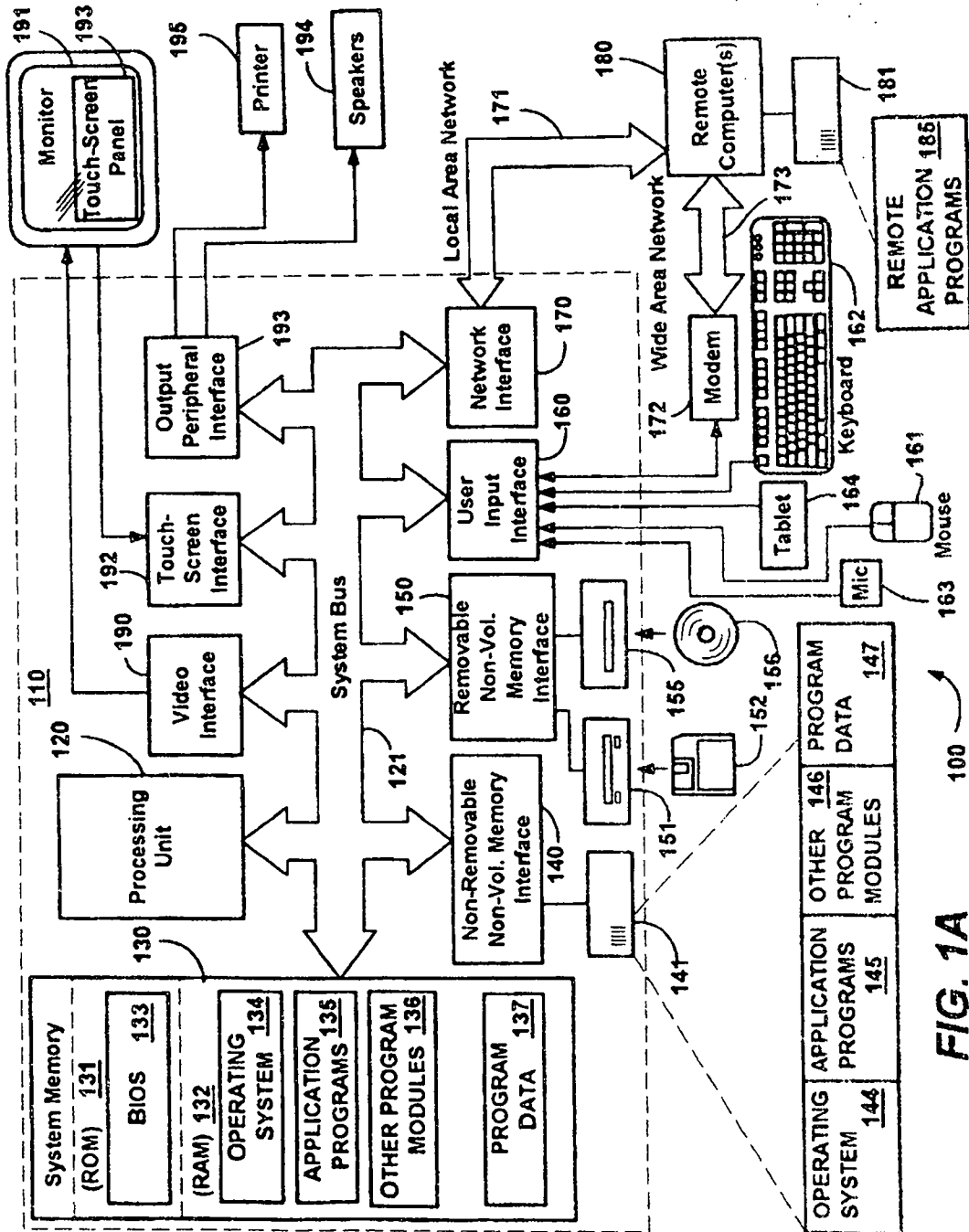


FIG. 1A

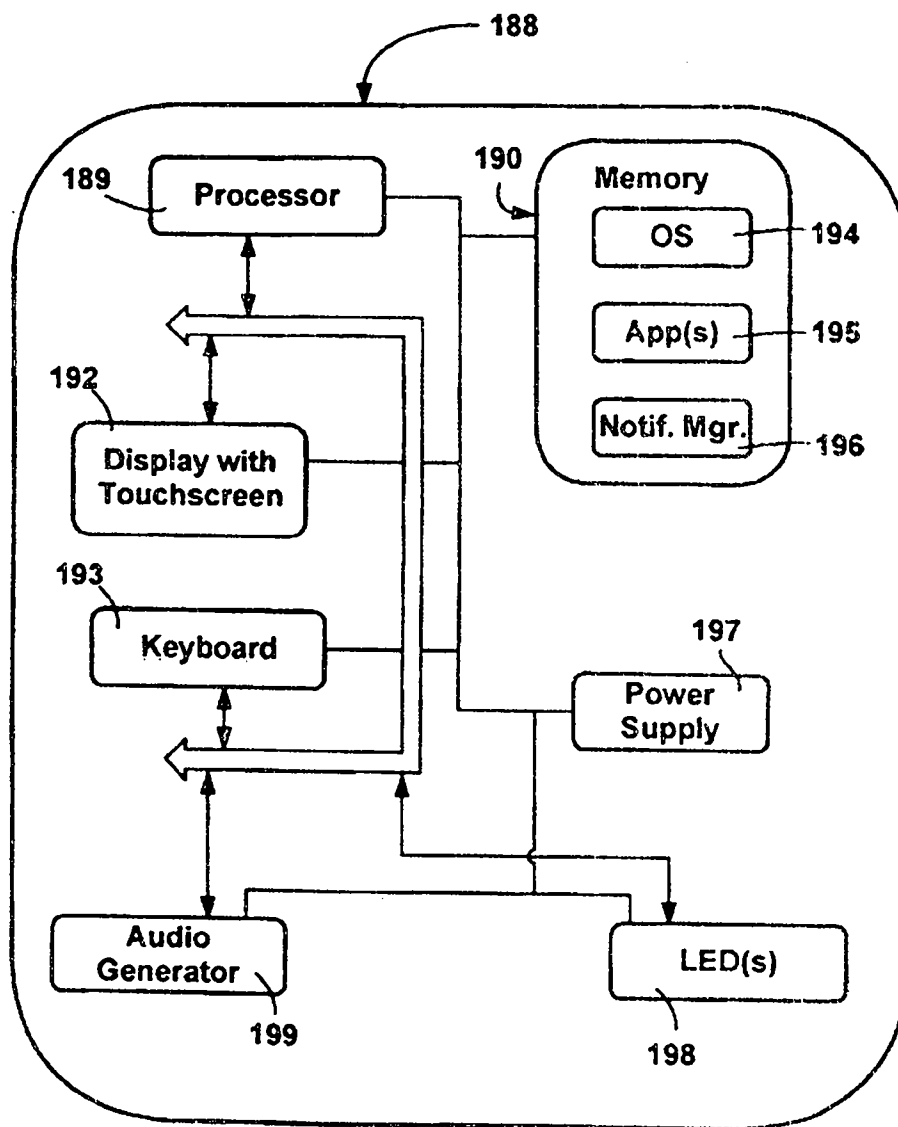


FIG. 1B

10187223.062802

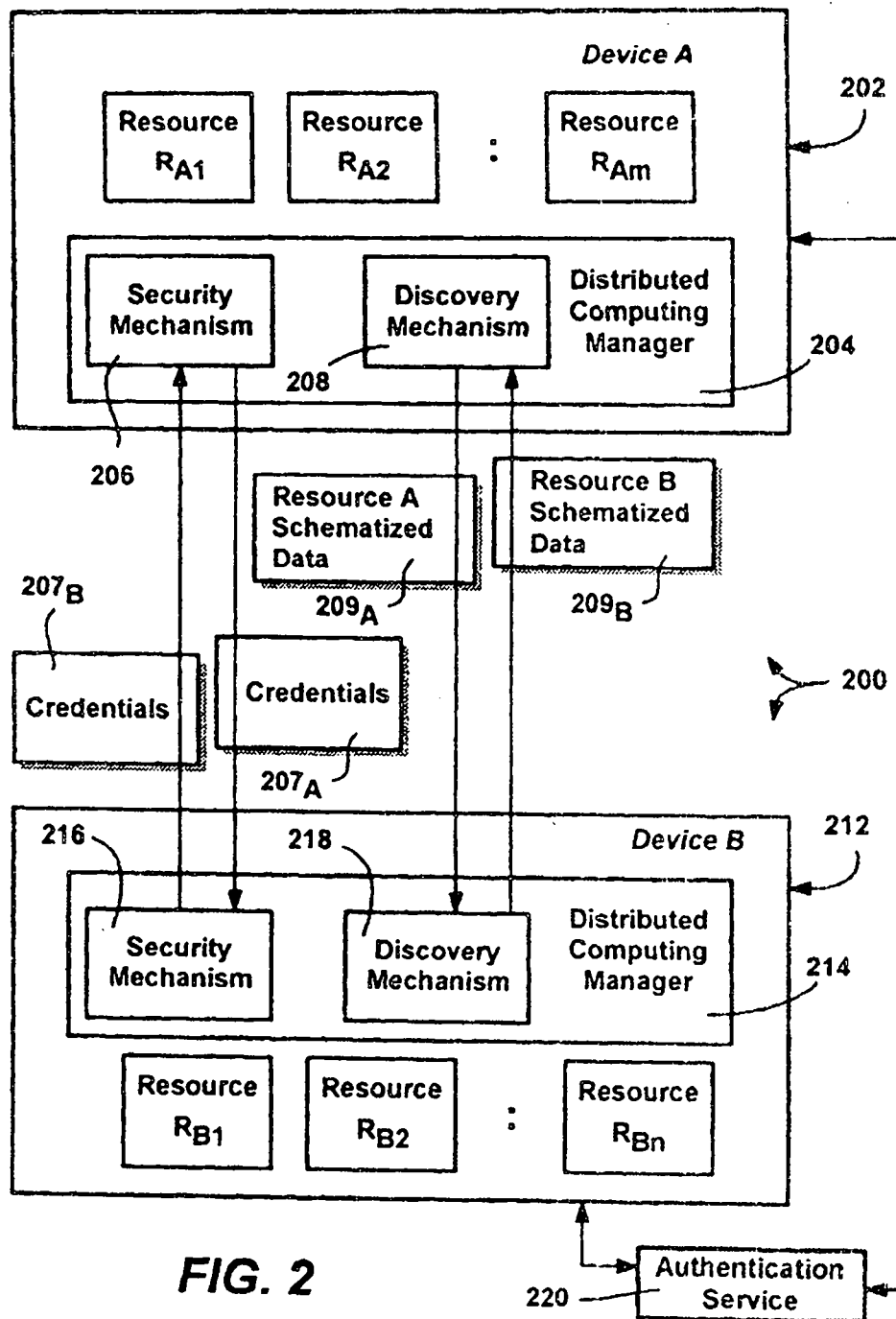


FIG. 2

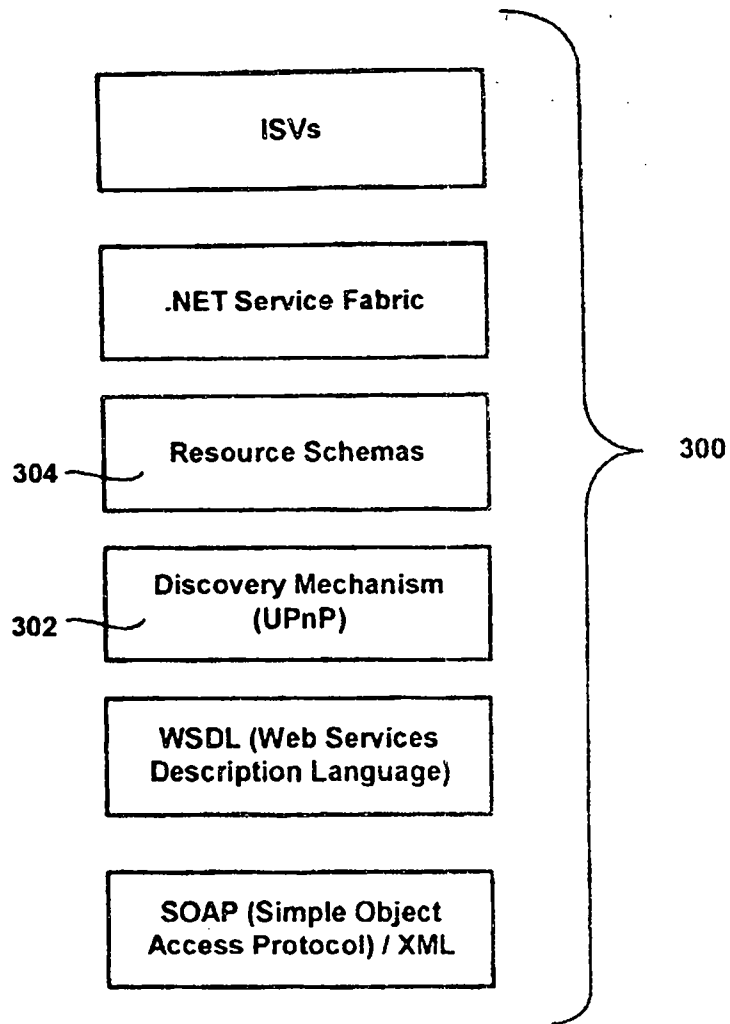


FIG. 3

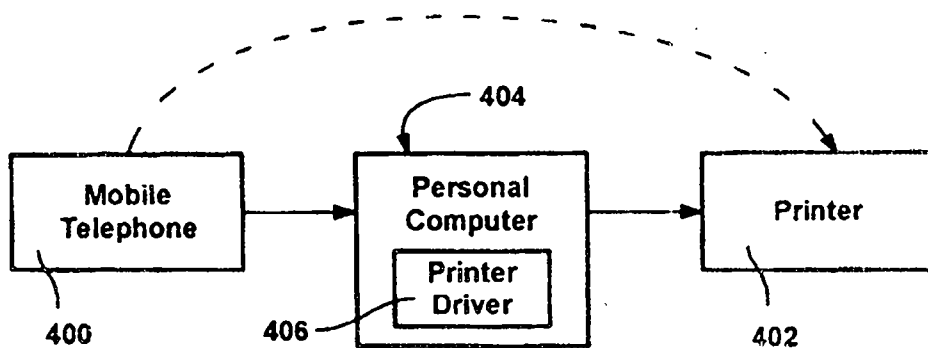


FIG. 4

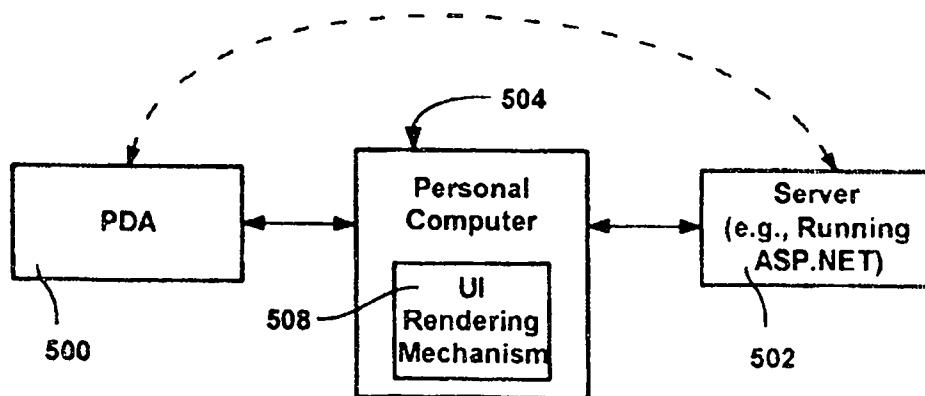
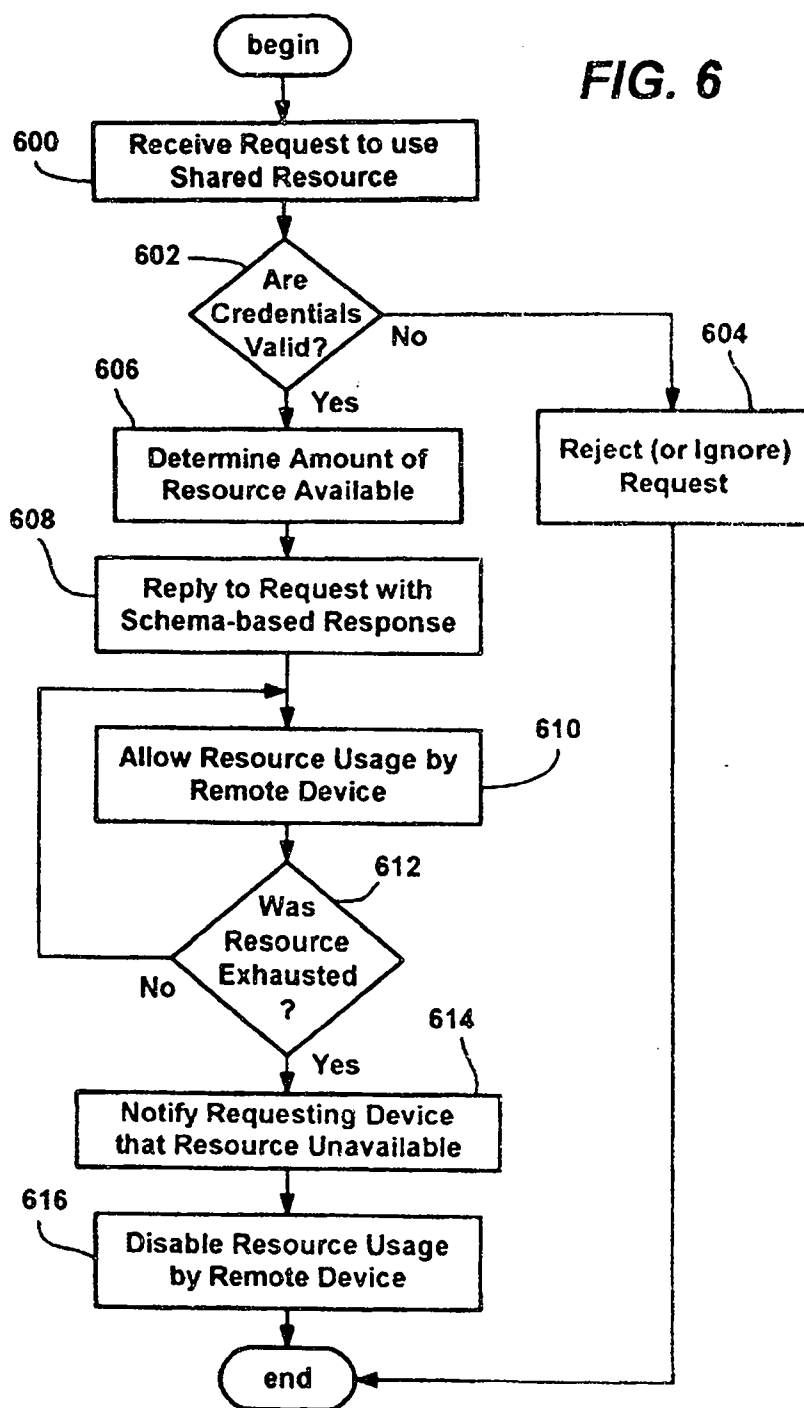


FIG. 5

FIG. 6



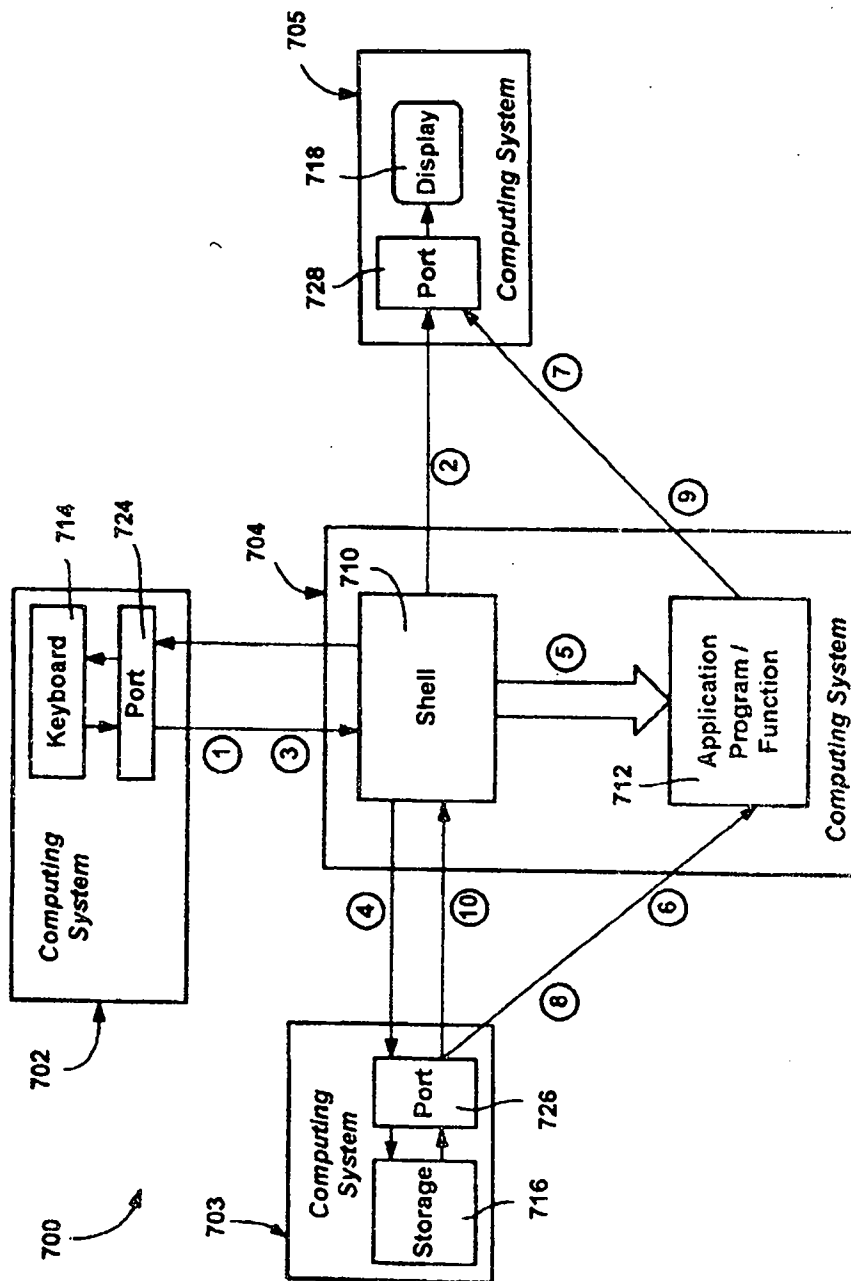


FIG. 7

737



P B 5818 - Patentlaan 2
2280 HV Rijswijk (ZH)
+31 70 340 2040
TX 3:65i epo nl
FAX +31 70 340 3016

Europäisches
Patentamt

Zweigstelle
in Den Haag
Eingangs-
stelle

European
Patent Office

Branch at
The Hague
Receiving
Section

Office européen
des brevets

Département à
La Haye
Section de
Dépôt

Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München
ALLEMAGNE



Datum/Date

11-06-2003

Zeichen/Ref./Réf.

EP27314-034/df

Anmeldung Nr./Application No./Demande n°./Patent Nr./Patent No./Brevet n°

03009717.4

Anmelder/Applicant/Demandeur/Patentinhaber/Proprietor/Titulaire

MICROSOFT CORPORATION

NOTIFICATION PURSUANT TO RULE 42 (ARTICLE 91(2) EPC)
DEFICIENCIES CONCERNING DESIGNATION OF INVENTOR

Your attention is drawn to the following deficiency/deficiencies in relation to the designation of the inventor in the above-mentioned European patent application.

() Where the applicant is the sole inventor: The designation is missing in the request for grant (Rule 26(2)(k) EPC).

- ☒ Where the applicant is not the inventor or the sole inventor:
- ☒ A separate document containing the designation drawn up according to Rule 17(1) and Article 81 EPC has not been filed.
 - () The document containing the designation does not comply with Article 81 and Rule 17(1) EPC since
 - () the document is not signed.
 - () the document is not duly signed. Reasons:

You are invited to remedy the deficiency/deficiencies:

- ☒ within 16 MONTHS after the date of filing or, if priority is claimed, after the earliest date of priority.
 - () within TWO MONTHS after notification of this invitation.
- These time limits are non-extendable.

If you fail to do so, the application will be deemed to be withdrawn (Art. 91(5) EPC).

RECEIVING SECTION



REGISTERED LETTER

EPO Form 1045 (03.00)

03009717.4 (FORI)

7002007 06/05/03

000

411

738



Antrag auf Erteilung eines europäischen Patents / Request for grant of a European patent / Requête en délivrance d'un brevet européen

1

Bestätigung einer bereits durch Telefax eingereichten Anmeldung / Confirmation of an application already
filed by facsimile / Confirmation d'une demande déjà déposée par téléfax
Wenn ja, Datum der Übermittlung des Telefax und Name der Einreichungsbehörde / If yes, facsimile date and name
of the authority with which the documents were filed / Si oui, date d'envoi du téléfax et nom de l'autorité de dépôt

☐ Ja / Yes / Oui

Datum / Date

Behörde / Authority / Autorité

Nur für amtlichen Gebrauch / For official use only / Cadre réservé à l'administration			
Anmeldenummer / Application No. / N° de la demande	MKEY	1	03009717.4
Tag des Eingangs (Regel 24(2)) / Date of receipt (Rule 24(2)) / Date de réception (règle 24(2))	DREC	2	30.04.03
Tag des Eingangs beim EPA (Regel 24(4)) / Date of receipt at EPO (Rule 24(4)) / Date de réception à l'OEB (règle 24(4))	RENA	3	
Anmeldetag / Date of filing / Date de dépôt		4	
Tabulatoren-Positionen / Tabulation marks / Arrêts de tabulation			
Es wird die Erteilung eines europäischen Patents und gemäß Artikel 94 die Prüfung der Anmeldung beantragt / Grant of a European patent, and examination of the application under Article 94, are hereby requested / Il est demandé la délivrance d'un brevet européen et, conformément à l'article 94, l'examen de la demande	EXAM 4	5	☒ Prüfungsantrag in einer zugelassenen Nichtamtssprache (siehe Merkblatt II, 5) / Request for examination in an admissible non-EPO language (see Notes II, 5) / Requête en examen dans une langue non officielle autorisée (voir notice II, 5)
Zeichen des Anmelders oder Vertreters (max. 15 Positionen) / Applicant's or representative's reference (maximum 15 spaces) / Référence du demandeur ou du mandataire (max. 15 caractères ou espaces)	AREF	6	EP27314-034/df
Anmelder / Applicant / Demandeur		7	MICROSOFT CORPORATION
Name / Nom		8	One Microsoft Way
Anschrift / Address / Adresse			Redmond, Washington 98052
			U. S. A.
APPR 01 # 7498660			
# DEST #			
Zustellanschrift / Address for correspondence / Adresse pour la correspondance		9	
PADR			
Staat des Wohnsitzes oder Sitzes / State of residence or of principal place of business / Etat du domicile ou du siège		10	U.S.A.
Staatsangehörigkeit / Nationality / Nationalité		11	U.S.A.
Telefon / Telephone / Téléphone		12	
Telex / Télex		13	
Telefax / Fax / Téléfax		14	
Weiterer(r) Anmelder auf Zusatzblatt / Additional applicant(s) on additional sheet / Autre(s) demandeur(s) sur feuille additionnelle		15	Grünecker, Kinkeldey, Stockmair & Schwanhäusser Anwaltssozietät
Vertreter / Representative / Mandataire		16	Zusammenschluß Nr. 72 / Association No. 72
Name / Nom			Maximilianstraße 58
Nur einen Vertreter angeben, der in das Europäische Patentregister eingetragen ist und an den zugestellt wird / Name only one representative who is to be listed in the Register of European Patents and to whom notification is to be made / N'indiquer qu'un seul manda- taire, qui sera inscrit au Registre européen des brevets et auquel signification sera faite			D-80538 München
FREP 01 1007210			Deutschland / Germany
Geschäftsanschrift / Address of place of business / Adresse professionnelle		17	+49 89 212350
Telefon / Telephone / Téléphone		18	+49 89 220287
Telex / Télex		19	
Telefax / Fax / Téléfax			
Weiterer(r) Vertreter auf Zusatzblatt / Additional representative(s) on additional sheet / Autre(s) mandataire(s) sur feuille additionnelle			

EPA/EPO/OEB Form 100; 1 07.02

TRAN ☐ FILL ☒ EM

EP27314-034/df
Raum für Zeichen des Anmelders / Space for applicant's
reference / Espace réservé à la référence du demandeur

1

412

Vollmacht / Authorisation / Pouvoir ist beigefügt / is enclosed / joint		20	☐
ist registriert unter Nummer / has been registered under No. / a été enregistré sous le n°		21	☐ Nummer Number Numéro
Erfinder / Inventor / Inventeur Anmelder ist (sind) alleinige(r) Erfinder / The applicant(s) is (are) the sole inventor(s) / Le(s) demandeur(s) est (sont) le (les) seul(s) inventeur(s) Erfindernennung in gesondertem Schriftstück / Designation of inventor attached / Voir la désignation de l'inventeur ci-jointe		22	☐
Bezeichnung der Erfindung / Title of invention / Titre de l'invention		23	☒
TIDE TIEN TIFR		24	IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES
Prioritätserklärung / Declaration of priority / Déclaration de priorité		25	Staat / State / Etat Anmeldetag / Date of filing / Date de dépôt Aktenzeichen / Application No. / N° de la demande
01 # # # 02 # # # 03 # # # 04 # # # Weitere Prioritätserklärungen) auf Zusatzblatt / Additional declaration(s) of priority on additional sheet / Autres) déclaration(s) de priorité sur feuille additionnelle		1 U.S.A. June 28, 2002 10/187,221 2 3 4	
Es wird hiermit erklärt, daß die Anmeldung eine vollständige Übersetzung der früheren Anmeldung ist (Regel 38(5)) / It is hereby declared that the application is a complete translation of the previous application (Rule 38(5)) / Il est déclaré par la présente que la demande est une traduction intégrale de la demande antérieure (règle 38(5))		25a	☐
Biologisches Material Die Erfindung bezieht sich auf bzw. verwendet biologisches Material, das nach Regel 28 hinterlegt worden ist. Die Angaben nach Regel 28(1)(c) (falls noch nicht bekannt, die Hinterlegungs- stelle und das (die) Bezugs- zeichen (Nummer, Symbole usw.) des Hinterlegers) sind in den technischen Anmeldungsunterlagen enthalten auf / The particulars referred to in Rule 28(1)(c) (if not yet known, the depository institution and the identification reference(s) (number, symbols etc.) of the depositor) are given in the technical documents in the application on / Les indications visées à la règle 28(1)(c) (si pas encore connues, l'autorité de dépôt et la (les) référence(s) d'identification (numéro ou symboles etc.) du déposant) figurent dans les pièces techniques de la demande à la /aux werden später mitgeteilt / will be submitted later /seront communiquées ultérieurement Die Empfangsbescheinigung(en) der Hinterlegungsstelle ist (sind) beigefügt / The receipt(s) of deposit issued by the depository institution is (are) enclosed / Le(s) récépissé(s) de dépôt délivré(s) par l'autorité de dépôt est (sont) joint(s) wird (werden) nachgereicht / will be filed later /sera (seront) produit(s) ultérieurement		26	Matière biologique L'invention concerne et/ou utilise de la matière biologique, déposée conformément à la règle 28.
BIO M : # # # # #		27	Seite(n) / page(s) Zeile(n) / line(s) / ligne(s)
		27a	☐
		27b	☐
		27c	☐

Falls das biologische Material nicht vom Anmelder, sondern von einem Dritten hinterlegt wurde: / Where the biological material has been deposited by a person other than the applicant: / Lorsque la matière biologique a été déposée par une personne autre que le demandeur: Ermächtigung nach Regel 28(1)(d) / Authorisation under Rule 28(1)(d) / L'autorisation en vertu de la règle 28(1)(d) ist beigelegt / is enclosed / est jointe wird nachgereicht / will be filed later / sera produite ultérieurement	28 Name und Anschrift des Hinterlegers / Name and address of depositor / Nom et adresse du déposant: 28a ☐ 28b ☐
Verzicht auf die Verpflichtung des Antragstellers nach Regel 28(3) in gesondertem Schriftstück / Waiver of the right to an undertaking from the requester pursuant to Rule 28(3) attached	29 ☐ Renonciation, sur document distinct, à l'engagement du requérant au titre de la règle 28(3)
Gemäß Regel 28(4) wird hiermit mitgeteilt, daß der Zugang zu dem in den Feldern 26 und 27 genannten biologischen Material nur durch Herausgabe einer Probe an einen Sachverständigen hergestellt wird / It is hereby declared under Rule 28(4) that the availability of the biological material referred to in Sections 26 and 27 shall be effected only by the issue of a sample to an expert BIOM 3	30 ☐ Conformément à la règle 28(4) il est déclaré par la présente que l'accèsibilité à la matière biologique mentionnée aux rubriques 26 et 27 ne peut être réalisée que par la remise d'un échantillon à un expert
Nucleotid- und Aminosäuresequenzen / Nucleotide and amino acid sequences / Séquences de nucléotides et d'acides aminés Die Beschreibung enthält ein Sequenzprotokoll nach Regel 27a(1) / The description contains a sequence listing in accordance with Rule 27a(1) Der vorgeschriebene Datenträger ist beigelegt / The prescribed data carrier is enclosed Es wird hiermit erklärt, daß die auf dem Datenträger gespeicherte Information mit dem schriftlichen Sequenzprotokoll übereinstimmt (Regel 27a(2)) / It is hereby stated that the information recorded on the data carrier is identical to the written sequence listing (Rule 27a(2)) SEOL 1	31 ☐ La description contient une liste de séquences selon la règle 27bis(1) ☐ Le support de données prescrit est joint ☐ Il est déclaré par la présente que l'information figurant sur le support de données est identique à celle qui contient la liste de séquences écrite (règle 27bis(2))
Benennung der Vertragsstaaten und Erklärungen hierzu 1. Hiermit werden sämtliche Vertragsstaaten des EPC benannt, die diesem bei Einreichung dieser Anmeldung angehören*. 2a. Es ist derzeit beabsichtigt, den siebenfachen Betrag einer Benennungsgebühr zu entrichten. Damit gelten die Benennungsgebühren für alle Vertragsstaaten als entrichtet (Art. 2 Nr. 3 GebO). 2b. Abweichend von der Erklärung in Nr. 2a ist derzeit beabsichtigt, weniger als sieben Benennungsgebühren für folgende Vertragsstaaten zu entrichten (bitte Ländercodes und Vertragsstaaten angeben*): (1) ☐ (2) ☐ (3) ☐	Désignation d'Etats contractants et déclarations à ce propos 1. Sont désignés tous les Etats qui sont des Etats contractants de la CEE à la date du dépôt de la présente demande*. 2a. Il est actuellement envisagé de payer un montant correspondant à sept fois la taxe de désignation. Les taxes de désignation sont ainsi réputées payées pour tous les Etats contractants (art. 2, point 3 du RR1). 2b. Contrairement à ce qui est indiqué au n° 2a, il est actuellement envisagé de payer moins de sept taxes de désignation pour les Etats contractants suivants (prière d'indiquer codes de pays et Etats contractants*): (4) ☐ (5) ☐ (6) ☐
Es wird beantragt, für die unter Nr. 2b nicht aufgeführten Vertragsstaaten von der Zustellung von Mitteilungen nach Regel 85a(1) und Regel 69(1) abzusehen. 3. Wird ein automatischer Abbuchungsauftrag erteilt (Feld 43), so wird das EPA beauftragt, bei Ablauf der Grundfrist nach Artikel 79(2) den siebenfachen Betrag einer Benennungsgebühr abzubuchen. Ist eine Erklärung unter Nr. 2b abgegeben worden, so sollen die Benennungsgebühren nur für die dort angegebenen Vertragsstaaten abgebucht werden, sofern dem EPA nicht bis zum Ablauf der Grundfrist ein anderslautender Auftrag zugeht.	Prière de ne pas procéder à la signification des notifications prévues par les règles 85bis(1) et 69(1) pour les Etats contractants n'ayant pas été mentionnés au n° 2b. 3. Si un ordre de prélèvement automatique est donné (rubrique 43), il est demandé à l'OEB de prélever, à l'expiration du délai normal visé à l'article 79(2), un montant correspondant à sept fois la taxe de désignation. Si une déclaration a été faite au n° 2b, les taxes de désignation ne sont prélevées que pour les Etats contractants qui y sont indiqués, sauf instruction contraire reçue par l'OEB avant l'expiration du délai normal.

* Stand bei Drucklegung: 24 Vertragsstaaten, und zwar: / Status when this form was printed: 24 contracting states, namely / Situation à la date d'impression: 24 Etats contractants, à savoir: AT Österreich / Austria / Autriche, BE Belgien / Belgium / Belgique, BG Bulgarien / Bulgaria / Bulgarie, CH/Li Schweiz und Liechtenstein / Switzerland and Liechtenstein / Suisse et Liechtenstein, CY Zypern / Cyprus / Chypre, CZ Tschechische Republik / Czech Republic / République tchèque, DE Deutschland / Germany / Allemagne, DK Dänemark / Denmark / Danemark, EE Estland / Estonia / Estonie, ES Spanien / Spain / Espagne, FI Finnland / Finland / Finlande, FR Frankreich / France / France, GB Vereinigtes Königreich / United Kingdom / Royaume-Uni, GR Griechenland / Greece / Grèce, IE Irland / Ireland / Irlande, IT Italien / Italy / Italie, LU Luxemburg / Luxembourg / Luxembourg, MC Monaco / Monaco / Monaco, NL Niederlande / Netherlands / Pays-Bas, PT Portugal / Portugal / Portugal, SE Schweden / Sweden / Suède, SK Slowakische Republik / Slovak Republic / République slovaque, TR Türkei / Turkey / Turquie

Verschiedene Anmelder für verschiedene Vertragsstaaten / Different applicants for different contracting states / Différents demandeurs pour différents Etats contractants		33 Name(n) des (der) Anmelder(s) und benannte Vertragsstaaten / Name(s) of applicant(s) and designated contracting states / Nom(s) du (des) demandeur(s) et des Etats contractants désignés	
APPR 02 #			
Erstreckung des europäischen Patents Diese Anmeldung gilt als Antrag, die europäische Patentanmeldung und das darauf erteilte europäische Patent auf alle Nicht-Vertragsstaaten des EPÜ zu erstrecken, mit denen am Tag ihrer Einreichung „Erstreckungsabkommen“ bestehen (derzeit: Albanien, Litauen, Lettland, Rumänien, Slowenien, ehemalige jugoslawische Republik Mazedonien). Die Erstreckung wird jedoch nur wirksam, wenn die vorgeschriebene Erstreckungsgebühr entrichtet wird.		34 Extension des effets du brevet européen ☒ La présente demande est réputée constituer une requête en extension des effets de la demande de brevet européen et du brevet européen délivré sur la base de cette demande à tous les Etats non parties à la CBE avec lesquels il existe un « accord d'extension » à la date du dépôt de la demande (Situation actuelle : Albanie, Lituanie, Lettonie, Roumanie, Slovénie, ex-République yougoslave de Macédoine). Toutefois, l'extension ne produit ses effets que s'il est acquitté la taxe d'extension prescrite.	
Extension of the European patent This application is deemed to be a request to extend the European patent application and the European patent granted in respect of it to all non-contracting states to the EPC with which "extension agreements" exist on the date on which the application is filed (Present situation: Albania, Lithuania, Latvia, Romania, Slovenia, former Yugoslav Republic of Macedonia). However, the extension only takes effect if the prescribed extension fee is paid.			
Es ist derzeit beabsichtigt, die Erstreckungsgebühr für die nachfolgend angekreuzten Staaten zu entrichten: / It is currently intended to pay the extension fee for the states marked below with a cross: / Il est actuellement envisagé de payer la taxe d'extension pour les Etats dont le nom est coché ci-après:			
Albanien / Albania / Albanie		AL	
Litauen / Lithuania / Lituanie		LT	
Lettland / Latvia / Lettonie		LV	
Rumänien / Romania / Roumanie		RO	
Slowenien / Slovenia / Slovénie		SI	
Ehemalige jugoslawische Republik Mazedonien / Former Yugoslav Republic of Macedonia / Ex-République yougoslave de Macédoine		MK	
(Platz für Staaten, mit denen nach Drucklegung dieses Formblatts „Erstreckungsabkommen“ in Kraft treten) / (Space for states with which "extension agreements" enter into force after this form has been printed) / (Place pour des Etats à l'égard desquels des «accords d'extension» entrent en vigueur après l'impression du présent formulaire)			
Die Anmeldung ist eine Teilanmeldung / The application is a divisional application / La présente demande constitue une demande divisionnaire		35 Nummer der früheren Anmeldung / No. of earlier application / Numéro de la demande initiale	
DFIL 9			
PANR			
Es handelt sich um eine Anmeldung nach Artikel 61(1)(b) / The application is an Article 61(1)(b) application / La présente demande constitue une demande selon l'article 61(1)(b)		36 Nummer der früheren Anmeldung / No. of earlier application / Numéro de la demande initiale	
DFIL 9			
EANR			
Patentansprüche / Claims / Revendications		37 29 Zahl der Patentansprüche / Number of claims / Nombre de revendications	
CLMS			
Zur Veröffentlichung mit der Zusammenfassung wird vorgeschlagen Abbildung Nr. / It is proposed that the abstract be published together with figure No. / Il est proposé de publier avec l'abrégé la figure n°		39 1A Nummer / Number / Numéro	
DRAW 2			

Zusätzliche Abschrift(en) der im europäischen Recherchenbericht angeführten Schriftstücke wird (werden) beantragt / Additional copy(ies) of the document(s) cited in the European search report is (are) requested / Prière de fournir une (des) copie(s) supplémentaire(s) des documents cités dans le rapport de recherche européenne ASOC	40 1 Anzahl der zusätzlichen Sätze von Abschriften / Number of additional sets of copies / Nombre de jeux supplémentaires de copies
Es wird die Rückerstattung der Recherchegebühr gemäß Art. 10 GebO beantragt / Refund of the search fee is requested pursuant to Article 10 of the Rules relating to Fees / Le remboursement de la taxe de recherche est demandé en vertu de l'article 10 du règlement relatif aux taxes Eine Kopie des Recherchenberichts ist beigelegt / A copy of the search report is attached / Une copie du rapport de recherche est jointe	41 42
Automatischer Abbuchungsauftrag (nur möglich für Inhaber von beim EPA geführten laufenden Konten) Das EPA wird hiermit beauftragt, fällig werdende Gebühren und Auslagen nach Maßgabe der Vorschriften über das automatische Abbuchungsverfahren vom nebenstehenden laufenden Konto abzubuchen. In bezug auf die Benennungsgebühren wird auf Feld 32.3 verwiesen. Das EPA wird ferner beauftragt, die Erstreckungsgebühren für jeden in Feld 34 angekreuzten «Erstreckungsstaat» bei Ablauf der Grundfrist zu ihrer Zahlung abzubuchen, sofern ihm nicht bis dahin ein anders lautender Auftrag zugeht. Für automatischen Abbuchungsauftrag: For automatic debit order: Pour l'ordre de prélèvement automatique: DECA	Automatic debit order (for EPO deposit account holders only) The EPO is hereby authorised, under the Arrangements for the automatic debiting procedure, to debit from the deposit account opposite any fees and costs falling due. With regard to designation fees reference is made to Section 32.3. The EPO is also authorised, on expiry of the basic period for its payment, to debit the extension fee for each of the "extension states" marked with a cross in Section 34, unless it is instructed to do otherwise before expiry of this period.
Eventuelle Rückzahlungen auf das nebenstehende beim EPA geführte laufende Konto / Any reimbursement to EPO deposit account opposite / Remboursements éventuels à effectuer sur le compte courant ci-contre ouvert auprès de l'OEB DEPA	Ordre de prélèvement automatique (possibilité offerte uniquement aux titulaires de comptes courants ouverts auprès de l'OEB) Par la présente, il est demandé à l'OEB de prélever du compte courant ci-dessous les taxes et frais venant à échéance, conformément à la réglementation relative à la procédure de prélèvement automatique. Pour les taxes de désignation, se reporter à la rubrique 32.3. Il est en outre demandé à l'OEB de prélever, à l'expiration du délai normal prévu pour leur paiement, les taxes d'extension pour chaque «Etat autorisant l'extension» coché à la rubrique 34, sauf instruction contraire reçue avant l'expiration de ce délai. Nummer des laufenden Kontos / Deposit account number / Numéro du compte courant Name des Kontoinhabers / Account holder's name / Nom du titulaire du compte
Die vorgeschriebene Liste über die diesem Antrag beigelegten Unterlagen ergibt sich aus der vorbereiteten Empfangsbescheinigung (Seite 6 dieses Antrages) The prescribed list of documents enclosed with this request is shown on the prepared receipt (page 6 of this request)	44 28 0004 37 45 La liste prescrite des documents joints à cette requête figure sur le récépissé préalable (page 6 de la présente requête)
Unterschrift(en) des (der) Anmelders(s) oder Vertreter(s) / Signature(s) of applicant(s) or representative(s) / Signature(s) du (des) demandeur(s) ou du (des) mandataire(s) Ort / Place / Lieu <u>München/Munich,</u> Datum / Date <u>April 30, 2003</u>	46 Für Angestellte nach Artikel 133(3) Satz 1 mit allgemeiner Vollmacht / For employees under Article 133(3), 1st sentence, having a general authorisation / Pour les employés mentionnés à l'article 133(3), 1^{re} phrase, munis d'un pouvoir général Nr. / No. / n°:

GRÜNECKER, KINKELDEY, STOCKMAIR & SCHWANHÄUSSER ANWALTSSOZIENTAT

Thomas Schuster

Name des (der) Unterzeichneten bitte in Druckschrift wiederholen. Bei juristischen Personen bitte die Stellung des (der) Unterzeichneten innerhalb der Gesellschaft in Druckschrift angeben. / Please print name under signature, in the case of legal persons, the position of the signatory within the company should also be printed. / Le ou les noms des signataires doivent être indiqués en caractères d'imprimerie. S'il s'agit d'une personne morale, la position occupée au sein de celle-ci par le ou les signataires doit être indiquée en caractères d'imprimerie.

743

Empfangsbescheinigung / Receipt for documents / Récépissé de documents 6

(Liste der diesem Antrag beigefügten Unterlagen) (Checklist of enclosed documents) (Liste des documents annexés à la présente requête)
 Es wird hiermit der Empfang der unten bezeichneten Dokumente bescheinigt / Receipt of the documents indicated below is hereby acknowledged / Nous attestons le dépôt des documents désignés ci-dessous

Wird im Falle der Einreichung der europäischen Patentanmeldung bei einer nationalen Behörde diese Empfangsbescheinigung vom Europäischen Patentamt übersandt, so ist sie als Mitteilung gemäß Regel 24(4) anzusehen (siehe Feld RENA). Nach Erhalt der Mitteilung nach Regel 24(4) sind alle weiteren Unterlagen, die die Anmeldung betreffen, nur noch unmittelbar beim EPA einzureichen. / If this receipt is issued by the European Patent Office and the European patent application was filed with a national authority it serves as a communication under Rule 24(4) (see Section RENA). Once the communication under Rule 24(4) has been received, all further documents relating to the application must be sent directly to the European Patent Office. / Si, en cas de dépôt de la demande de brevet européen auprès d'un service national, l'Office européen des brevets délivre le présent récépissé de documents, ce récépissé est réputé être la notification visée à la règle 24(4) (cf. rubrique RENA). Dès que la notification visée à la règle 24(4) a été reçue, tous les autres documents relatifs à la demande doivent être adressés directement à l'OEB.

Grünecker, Kinkeldey, Stockmair
 & Schwanhäusser
 Anwaltssozietät
 Maximilianstraße 58
 D-80538 München

Nur für amtlichen Gebrauch / For official use only / Cadre réservé à l'administration

Datum / Date

 Europäisches Patentamt
 European Patent Office
 Office européen des brevets
 D-80298 München
 A. Alders-Meewis

Anmeldenummer / Application No. / N° de la demande		83009717.4	
Tag des Eingangs (Regel 24(2)) / Date of receipt (Rule 24(2)) / Date de réception (règle 24(2))	DREC	30.04.03	
Zeichen des Anmelders/Vertreters / Applicant's/ Representative's ref. / Référence du demandeur ou du mandataire	AREF		
Nur nach Einreichung der Anmeldung bei einer nationalen Behörde: / Only after filing of the application with a national authority: / Seulement après le dépôt de la demande auprès d'un service national:			
Tag des Eingangs beim EPA (Regel 24(4)) / Date of receipt at EPO (Rule 24(4)) / Date de réception à l'OEB (règle 24(4))	RENA		
A. Anwendungsunterlagen und Prioritätsbeleg(e) / Application documents and priority document(s) / Pièces de la demande et document(s) de priorité		47	
1. Beschreibung (ohne Sequenzprotokoll(e)) / Description (excluding sequence listing part) / Description (sans partie réservée au listage des séquences)		☒	Blattzahl* / Number of sheets* / Nombre de feuilles* 40
2. Patentansprüche / Claims / Revendications		☒	7
3. Zeichnungen / Drawings / Dessins		☒	7
4. Sequenzprotokoll(e) der Beschreibung / Sequence listing part of description / Partie de la description réservée au listage des séquences		☒	1
5. Zusammenfassung / Abstract / Abrégé		☒	
6. Übersetzung der Anwendungsunterlagen / Translation of the application documents / Traduction des pièces de la demande		☐	
7. Prioritätsbeleg(e) / Priority document(s) / Document(s) de priorité		☐	
8. Übersetzung des (der) Prioritätsbeleg(s) / Translation of priority document(s) / Traduction du (des) document(s) de priorité		☐	
B. Der Anmeldung in der eingereichten Fassung liegen folgende Unterlagen bei: / This application as filed is accompanied by the items below: / A la présente demande sont annexés les pièces suivantes:		48	
1. Einzelvollmacht / Specific authorisation / Pouvoir particulier		☐	
2. Allgemeine Vollmacht / General authorisation / Pouvoir général		☐	
3. Erfindernennung / Designation of inventor / Désignation de l'inventeur		☐	
4. Früherer Recherchenbericht / Earlier search report / Rapport de recherche antérieur		☐	
5. Gebührenzahlungsvordruck (EPA Form 1010) / Voucher for the settlement of fees (EPO Form 1010) / Bordereau de règlement de taxes (OEB Form 1010)		☒	
6. Scheck (nicht bei Einreichung bei den nationalen Behörden) / Cheque (not when filing with national authorities) / Chèque (pas de chèque en cas de dépôt auprès des services nationaux)		☐	
7. Datenträger für Sequenzprotokoll / Data carrier for sequence listing / Support de données pour liste de séquences		☐	
8. Zusatzblatt / Additional sheet / Feuille supplémentaire		☐	
9. Sonstige Unterlagen (bitte hier spezifizieren) / Other documents (please specify here) / Autres documents (veuillez préciser)		☐	
C. Kopien dieser Empfangsbescheinigung / Copies of this receipt for documents / Copies du présent récépissé de documents		49	
		2	Anzahl der Kopien / Number of copies / Nombre de copies

* Die Richtigkeit der Angabe der Blattzahl und der Gesamtzahl der Abbildungen wurde bei Eingang nicht geprüft / No check was made on receipt that the number of sheets and the total number of figures indicated were correct / L'exactitude du nombre de feuilles et du nombre total de figures n'a pas été contrôlée lors du dépôt

Zur Kasse
 1595,-

417

744

EPO - Munich
55
30. April 2003

EP27314-034/df

30.04.2003

MICROSOFT CORPORATION
One Microsoft Way
Redmond, Washington 98052
U. S. A.

**IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE
RESOURCES**

418

IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

FIELD OF THE INVENTION

The invention relates generally to computer systems, and
5 more particularly to computer resources.

BACKGROUND OF THE INVENTION

In early personal computing systems, application program
had to be programmed with knowledge of various computing
10 devices in order for the application to use such devices. For
example, in order to print to a particular printer, a word
processing program had to know how to print to the printer,
and generally supplied a driver for each supported printer.
Alternatively, a printer would supply drivers with it for the
15 most popular programs. The user had to configure each program
to inform that program as to which printer was connected to
the personal computer.

Contemporary operating systems such as Microsoft Windows®
provided developers with a way to interface applications with
20 such devices regardless of the specific device driver needs
and requirements. For example, with the Microsoft Windows®
operating system a user or the device tells the operating
system which printer (or printers) is connected, after which
any application is able to communicate with the printer driver
25 via the operating system. As a result, to accomplish

printing, application programs generally only need to deal with a set of standardized printing functions provided by the operating system. The enormous benefits of such an abstraction cannot be understated, e.g., this abstraction enables any one of thousands of software programs to interact with any one of thousands of printers and/or other devices, without the program having to know (or otherwise supply) the details about the printer in advance.

Microsoft Corporation's .NET technology provides a somewhat similar type of abstraction regarding a user's or company's data. In general, with .NET technology, any .NET-capable program on any device can interact with a user's data via centralized schema-based services. Because of the schematization, the programs do not need specific knowledge about the details of that data, e.g., which used to be stored in proprietary file formats, but rather only needs to be able to deal with (read and write) data formatted according to one or more of the schemas.

With .NET technology, a user's (or other entity's) identity controls access to the centrally-maintained data, e.g., a user may use a .NET Inbox service keep his or her electronic mail data centrally stored (e.g., on the Internet or an intranet), and then access that data via the .NET inbox service, at any time, from any device capable of connecting to

the Internet or an intranet and having a program capable of interacting with the .NET Inbox service. The user's identity is used to securely match the user with the appropriate data. Users can also controllably allow access to data by other
5 users, e.g., a user can selectively make certain business contact information available to business associates, make personal contact information available to friends, and so on, based on the others' identities.

While .NET thus provides tremendous data-based benefits
10 to users, companies and other entities, the centralized service and centralized data are not particularly concerned with how a user uses various devices and those devices' resources. For example, devices such as a personal computer and pocket personal computer have, for quite some time, been
15 able to directly synchronize with one another. To an extent, .NET technology is another way to keep such data synchronized, although indirectly, without needing specialized and proprietary synchronization programs.

However, even with .NET, a user cannot easily use one
20 device to access the resources of another device, such as to use the computing power, storage, bandwidth, input and output mechanisms, programs and/or other resources of one device on behalf of another device, e.g., to perform a complex or better-suited task on the more powerful device. What is

needed is a common, consistent way to remotely discover and expose computing resources of various computing devices to one another for cross-resource usage.

5

SUMMARY OF THE INVENTION

Briefly, the present invention provides a system and method for discovering and describing computing resources available at various computing devices associated with an identity, and for exposing those resources as services that are addressable by software programs, such that programs on the devices can leverage another device's resources. The device services are identity-based, in that a device owner's devices discover each other and/or other devices accessible by a user based on the owner's or other user's identity.

15 Further, based on identity, a device owner (or administrator) can control what other users, and/or other devices can leverage the resources of each owned device, as well as to what extent.

Computing resources may include, for example, the storage capacity of the devices, bandwidth, processing power (e.g., the CPU and RAM capabilities), input mechanisms and methods, rendering and other output mechanisms and methods (e.g., screens, printers and audio), and programs. A device is thus abstracted into its resources, which are then programmatically

addressable from other devices in a composable fashion (e.g.,
the remote resources may be defined in such a way that they
are, in turn, comprised of smaller resources contained within
the larger definition of the resource) . In other words, for
5 devices to participate in an intelligent distributed computing
model, various aspects of each participating device are
abstracted and made programmatically addressable from other
devices. A unified identity-based mechanism discovers the
list of remote resources on the device. A set of one or more
10 schemas consistently describes the remote resources on the
device, and a unified security model allows the owner of the
device to control who has access to which resources, and to
what extent.

With the present invention, desirable computing solutions
15 for a great number of example scenarios are enabled, including
a set of real-time scenarios in which one user device
interacts with one or more remote computing devices in real-
time, and a set of scheduled or predicted scenarios that
operate transparently and automatically on a user's behalf.

20 Other advantages will become apparent from the following
detailed description when taken in conjunction with the
drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1A is a block diagram generally representing a computer system into which the present invention may be incorporated;

5 FIG. 1B is a block diagram generally representing another computer system into which the present invention may be incorporated;

FIG. 2 is a block diagram representing various components in a distributed computing environment in accordance with an aspect of the present invention;

10 FIG. 3 is a block diagram generally representing an architecture for discovering and exchanging resource information in accordance with an aspect of the present invention;

15 FIGS. 4 and 5 are block diagrams representing examples of how the present invention can improve a user computing experience by distributing computing operations to resources, in accordance with an aspect of the present invention;

FIG. 6 is a flow diagram representing exemplary logic and related operations performed to facilitate the sharing of resources in a distributed computing environment in accordance with an aspect of the present invention; and

FIG. 7 is a block diagram generally representing an example implementation including resources distributed in accordance with an aspect of the present invention.

5

DETAILED DESCRIPTION

EXEMPLARY OPERATING ENVIRONMENTS

FIGURE 1A illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one
10 example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components
15 illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for
20 use with the invention include, but are not limited to: personal computers, server computers, hand-held or laptop devices, tablet devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe

computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, 5 being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth, that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where 10 tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in local and/or remote computer storage media including memory storage devices.

15 With reference to FIG. 1A, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 20 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and

not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component

5 Interconnect (PCI) bus also known as Mezzanine bus. Note that any and all of the elements shown in computer 110 may be connected to each other by standard high-speed computer networks, including those distributed across a wide geographic area. For example, the system memory 130 may be located in a
10 physically different location than processing unit 120, but yet defined in a composable fashion as logical computer 110.

The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 110 and
15 includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-
20 removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,

digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can

5 accessed by the computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data
10 signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media
15 such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read
20 only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or

program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, FIG. 1A illustrates operating system 134, application programs 135, other program modules 5 136 and program data 137.

The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1A illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile 10 magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, 15 volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is 20 typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

The drives and their associated computer storage media, discussed above and illustrated in FIG. 1A, provide storage of computer-readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1A, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as mouse, trackball or touch pad. Other input devices not shown in FIG. 1A may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected

to the system bus 121 via an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 195 and printer 196, which may be connected through an output peripheral interface 194 or the like.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1A. The logical connections depicted in FIG. 1A include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. For example, in the present invention, the computer system 110 may comprise source machine from which data is being migrated,

and the remote computer 180 may comprise the destination machine. Note however that source and destination machines need not be connected by a network or any other means, but instead, data may be migrated via any media capable of being
 5 written by the source platform and read by the destination platform or platforms.

When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the
 10 computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160 or other appropriate mechanism. In a networked
 15 environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1A illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network
 20 connections shown are exemplary and other means of establishing a communications link between the computers may be used.

FIGURE 1B illustrates an example of another suitable operating environment (computer device 188) in which at least

part of the present invention may be implemented. As with FIG. 1A, the operating environment of FIG. 1B is only one other example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. The other computer device 188 in FIG. 1B typically includes at least some form of computer readable media. Computer-readable media can be any available media (as generally defined above) that can be accessed by the computing device 188.

FIG. 1B shows functional components of a computing device 188 such as a handheld (pocket-sized or tablet) personal digital assistant, including a processor 189, a memory 190, a display 192, and a keyboard 193 (which may be a physical or virtual keyboard). The memory 190 generally includes both volatile memory (e.g., RAM) and non-volatile memory (e.g., ROM, PCMCIA cards, and so forth). An operating system 194 is resident in the memory 190 and executes on the processor 189, such as the Windows® CE operating system from Microsoft® Corporation, or another operating system.

One or more application programs 195 are loaded into memory 190 and run on the operating system 194. Examples of applications include email programs, scheduling programs, PIM (personal information management) programs, word processing programs, spreadsheet programs, Internet browser programs, and

so forth. The handheld personal computer 188 may also include
 a notification manager 196 loaded in the memory 190, which
 executes on the processor 189. The notification manager 196
 handles notification requests, e.g., from the application
 5 programs 195.

The handheld personal computer 188 has a power supply
 197, which is implemented as one or more batteries. The power
 supply 197 may further include an external power source that
 overrides or recharges the built-in batteries, such as an AC
 10 adapter or a powered docking cradle.

The exemplary handheld personal computer 188 represented
 in FIG. 1B is shown with three types of external notification
 mechanisms: one or more light emitting diodes (LEDs) 198 and
 an audio generator 199. These devices may be directly coupled
 15 to the power supply 197 so that when activated, they remain on
 for a duration dictated by a notification mechanism even
 though the handheld personal computer processor 189 and other
 components might shut down to conserve battery power. The LED
 198 preferably remains on indefinitely until the user takes
 20 action. Note that contemporary versions of the audio
 generator 199 use too much power for today's handheld personal
 computer batteries, and so it is configured to turn off when
 the rest of the system does or at some finite duration after
 activation.

IDENTITY-BASED DISTRIBUTED COMPUTING FOR DEVICE RESOURCES

The present invention is generally directed to a system and method that facilitate the distribution of resources of devices to other devices. The present invention is ordinarily not involved with what the devices do with each other's resources, what is communicated at that time, or how the communication between devices takes place. Rather, the present invention can be generally considered more as a structured way for devices to find out about each other's resources, capabilities and other details, and then allow the devices to share resources on their own. Although not necessary to the present invention, as part of facilitating the resource distribution, security is provided for practical purposes, at least to ensure that valid credentials are required to use another device's resources. Thus, the present invention is preferably identity-based, in that distributed resources have relationships with one another according to identity data (e.g., credentials) provided to the devices.

In general, the present invention is directed to distributing computing among a user's devices, wherein a user (or owner / administrator) is anyone authorized to control and/or use the devices. However, because in general devices are increasingly becoming disaggregated, the methods and

systems described herein may be more properly described as distributed computing of resources. For example, a single distributed device may be made up of many resources, but may share as little as one its resources (e.g., only its non-
5 volatile storage) to participate as a resource provide. Notwithstanding, as used herein, the terms "device" and "resource" are generally interchangeable except where otherwise apparent, e.g., using another device or using a resource of the other device to accomplish some computing task
10 are generally considered the same thing. Also, as used herein, bandwidth is a device resource, even though bandwidth may not ordinarily be considered as being associated with any particular device.

As generally represented in FIG. 2, a distributed
15 computing environment 200 constructed in accordance with various aspects of the present invention includes a first device 202 (Device A, such as the computer system 110 of FIG. 1A) and a second device 212 (Device B, such as the computer system 188 of FIG. 1B), along with possibly other devices (not
20 shown). Each device that participates in the distributed computing environment 200 includes a distributed computing manager, 204 and 214. In general, and as described below, the distributed computing managers 204 and 214 may comprise some amount of code that communicate with one another over some

medium, e.g., wireless FM, line-of-sight such as infrared, satellite, a direct connection, the Internet, or an intranet. Indeed, more than one such medium may be available, and the distributed computing managers 204 and 214 may choose which
5 one or ones are best for a given circumstance.

An initial purpose of the distributed computing managers 204 and 214 are to locate each other. Such location may be automatic, such as by having each device occasionally broadcast its availability to others and/or by polling for the
10 availability of other devices, or alternatively may operate in response to some specific request from a program (or user) to connect to another device. As represented in FIG. 2, the distributed computing managers 204 and 214 each include a security mechanism 206 and 216, respectively, that exchange
15 credentials 207, 217 or the like (e.g., obtained from an authentication service 220) to ensure that the user who is currently operating the device which is contacting it (and/or the device itself), is valid in the distributed computing environment, and similarly, if necessary, that the device
20 which is being contacted is valid. Note that a program being used may also factor into the security / access control, e.g., a first program running on one device may have authority to use another device's resource or resources, but not another program, independent of or in addition to the user identity

and/or the device identity. Further note that the security mechanism may operate per identity and per resource, e.g., one entity may be authorized to use a device's display, but not its storage, while another entity may be able to use any of that same device's available resources, including its storage. This is similar to the .NET security model in which services provide methods that control access to a users' data based on the requesting user's identity, defined role and scope for that role. In this way, data can be accessed by its owner, and shared to an extent determined by the owner. In the present invention, instead of data it is a device resource being accessed, whereby a user's identification, defined role and scope for that role determine the extent to which a resource may be accessed.

Indeed, the security mechanisms of the present invention are based on those essentially those employed for security / access control in .NET services, as generally described in United States patent applications, serial nos. 10/003,754 entitled "IDENTITY-CENTRIC DATA ACCESS" and serial number 10/003,767, entitled "AUTHORIZING A REQUESTING ENTITY TO OPERATE UPON DATA STRUCTURES" filed on October 22, 2001, assigned to the assignee of the present application, and herein incorporated by reference. To this end, at least a thin subset of the .NET service runs on each device which

expects to advertise and share computing resources in the environment 200. In addition, a localized internal consistent security model may be executed on these devices such that local security is provided even when two local peer-to-peer devices may be disconnected from the centralized (e.g., Passport) authentication service 220 (FIG. 2).

The present invention is not limited to any particular transmission medium, but provides that the distributed resource services may be directly addressable to the target device, or may be queried via the Internet or an intranet, such as via the centralized set of .NET services on the web. When connected via the Internet or intranet, a program may query the last known computing resources available on any device associated with the user's identity, even when the device is offline or otherwise inaccessible. Bandwidth is also a resource, and may impact how a device's resources are shared, e.g., it may be more efficient to leverage a personal computer's resources when the personal computer has a high-speed connection to the Internet, but not if the personal computer connects only at low speed, such as if the other device has a high-speed connection. In fact, the personal computer may want to leverage the high-speed connection of the other device, in which event that other device may then leverage a resource of the personal computer, e.g., the

devices exchange resources to provide a more-optimal user experience.

In accordance with one aspect of the present invention, the distributed computing managers 204 and 214 also each
5 include a respective discovery mechanism, 208 and 218, that allows the devices to discover another participating device's shared resources / capabilities, that is, when that other device is configured to share its resources. The respective resource data 209 and 219 is formatted (e.g., in XML) and
10 exchanged in accordance with a defined schema, as described below. As represented in FIG. 2 by the exchange of the schematized data 209 and 219, both devices 202 and 204 are configured to share their resources ($R_{A1}-R_{Am}$ and $R_{B1}-R_{Bn}$) with other devices in the distributed computing environment.
15 However as can be appreciated, in a given environment, a particular device (e.g., a mobile telephone) may be only a consumer of resources rather a provider of any resource, or conversely may be only a provider, and not a consumer.

A device that is sharing its resources need not directly
20 communicate the resource data with the other device, but instead, for example, can advertise its resources somewhere for other devices to discover. For example, a .NET service (.NET Devices) is provided that centrally stores device resource capabilities such that data may be appropriately

formatted before sending it to a device, e.g., so as to not
 send high-resolution graphics to a device incapable of
 displaying them. A similar .NET Resources service, or an
 extended .NET Devices service, may be provided to advertise
 5 this information to other devices.

The following tables set forth some example schemas that
 may be used (e.g., with XML-formatted data) to exchange
 resource (storage and processing power) information regarding
 a user's devices, which in this example include work and home
 10 personal computers, and a cellular telephone:

```

<myStorage>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <driveSize/>
    <freeSpace/>
    <transferSpeed/>
    <seekLatency/>
  </device>
</myStorage>
  
```

```

<myProcessingPower>
  <device name="Work PC" uuid="7a733d84-b394-4ba1-56b5c0d">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Home PC" uuid="6d733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
  <device name="Cell Phone" uuid="5f733d84-b3f4-4ba1-56b5c0f">
    <cpuSpeed/>
    <cpuUsage/>
    <memorySize/>
    <memoryUsage/>
  </device>
</myProcessingPower>

```

As is understood, the schema is preferably resource based, i.e., multiple devices may be listed for each resource type, to match the general concept of an identity having distributed resources available for use, independent of the actual device on which the resource exists. Notwithstanding, the present invention is not limited to any particular schema and/or formats, e.g., the schemas may be device-based, such that each device may instead provide its list of available resources to other devices such that the same information is known.

As generally represented in FIG. 3, an architecture 300 is provided in order for devices to participate in an intelligent distributed computing model, in which various

aspects of the participating devices are abstracted and made programmatically addressable from a different device. A universal plug-and-play (UPnP 2.0) infrastructure or the like may be used as the basic, low-level mechanism 302 for
5 discovery of these resources. Alternatively, the list of resources for various computing devices may be cached in some well-known location, such as an Active Directory store, to enable querying of remote resources even if the device is only accessible through some intermediary such as a firewall or
10 wireless hub. Atop the discovery mechanism are the schemas 304, e.g., one schema associated with each specific type of the computing resource, with each of those schemas exposed, for example, as in the .NET architecture.

Note that while UPnP provides a suitable discovery
15 mechanism 304, it is only part of the process of actually using the remote computing resources. Thus, UPnP and .NET technologies are used to facilitate resource sharing, but are not used to actually employ the resources available on a device. Rather, once the resources are known, appropriately
20 focused protocols for that specific resource domain are used. By way of example, UPnP, with the addition of the .NET identityHeader, endpointHeader, and requestHeader SOAP semantics, is well-suited to obtain the video screen characteristics of the remote device, however a protocol such

as RDP will be used to transfer video data after the initial discovery and handshaking occurs between the devices. Note that this is similar to the way .NET approaches data services today, e.g., .NET provides a mechanism to connect to an
5 electronic endpoint representing a proprietary (e.g., Messenger) protocol, but once that endpoint is connected, .NET allows the client to use an existing real-time protocol. Other protocols appropriate for other types of resources are used, e.g., CIFS+ for storage, which scales well to both
10 corporate and home storage scenarios and enables the user's file system to continuously expand to other storage devices, without forcing the user to deal with drive letters, namespaces, and so forth. In summary, the present invention provides an identity-based approach to discovering these
15 additional resources that are available for use, but does not thereafter directly deal with their use.

FIG. 6 is an example flowchart that represents some general aspects of the present invention, beginning with step 600 which represents receiving a request to use a shared
20 resource. The request may be general, e.g., requesting the list of resources that a user has available on devices configured to participate, or may be a request for a specific resource, e.g., in response to an advertisement. For purposes

of this example, the request is treated as a request for one specific resource.

Step 602 represents the evaluation of the credentials, such as provided by the Microsoft .NET Passport authentication service. The request may include the credentials, or the credentials may be separately exchanged, e.g., before, as part of, or after step 600. If not valid at step 602, step 602 branches to step 604 which represents rejecting the request. Note that depending on a given environment, the request without proper credentials may be ignored rather than specifically rejected.

If valid, step 606 is executed, and represents determining the amount of the requested resource that is available. For example, a resource such as storage may be limited because it has only limited free space, and further, a device may be configured to make only part of its storage space available for sharing, reserving the rest for itself. A resource such as processor cycles may be made available only to the extent the processor is not being otherwise used, and even then may be further limited so that another device can only take up to a maximum amount of the unused cycles. Any resource may have a time-out value associated with it, which may be further contingent upon its use, e.g., storage can be shared for no more than one hour per request, and the request

will be terminated if no file system requests are detected at least every two minutes. In this manner, a device may be configured to share its resources to an extent controlled by its owner / administrator.

5 Note that the resource access may be dynamic, e.g., unused processor cycles may be shared, however CPU usage is a number that fluctuates, and the requesting guest device may be prioritized along with other devices and/or processes on the device that is sharing the CPU. For example, guest users may
10 only consume a controllable percentage of resources when the owner of the device is not otherwise using them, such as bandwidth or processing power. Both CPU and bandwidth are examples of resources which are scheduled according to time constraints (e.g., cycles or bits per second) rather than non-
15 time-based resources such as disk or memory usage. As a result, the scheduling mechanisms for time-based resources in general will be fundamentally different than for non-time-based resources. Note that kiosk locations would enable many scenarios that would be useful to such guests. A resource
20 such as a display monitor may be either entirely available for sharing or not at all, depending on whether or not it is presently being used. Thus, although not represented in FIG. 6, step 606 may also reject (or ignore, if appropriate) a request when there is no amount of a requested resource

available for sharing. Step 608 represents returning a response to the requesting entity, preferably the schema-based response that provides the resource capability data to the requesting entity.

5 Step 610 represents allowing the requested resource to be shared, and includes, for example, the exchange of data between the requesting device and the sharing device so that the sharing performs some corresponding task for the requesting device. Step 612 tests whether the resource was
10 exhausted, e.g., timed out or is no longer available. Note that this is shown as a loop in FIG. 6 for purposes of simplicity, however it is readily understood that the actions may be event driven. Further, a step (such as step 606) may be included in the loop / used to fire the event, e.g., the
15 resource usage is dynamically monitored to determine when it is exhausted for the particular request. Note that a resource may be directly exhausted when the requesting entity informs the device that the resource is no longer needed.

Step 614 represents notifying the requesting device that
20 the resource is no longer available for its use. Step 616 represents disabling the resource usage, so that, for example, once exhausted the requesting device will need to make a new request if it wants to share resources.

FIGS. 4 and 5 provide examples of some of the many possible uses of the present invention. As will be understood from the examples of FIGS. 4 and 5, integrating computational proxy support into a network edge device or appliance may
5 result in a better user experience for users of an internal network.

In a first example represented in FIG. 4, a small mobile device 400 (such as a mobile telephone) wants to print to a printer 402, however the small device 400 does not have an
10 appropriate printer driver. This may be because the mobile device 400 does not have sufficient memory to install and maintain a printer driver needed for that printer 402, and/or does not have an easy way for the needed printer driver to be installed, and/or for other reasons. Note that this is a real
15 scenario, and one proposed solution is to emit XHTML-PRINT as a simplified PDL (Page Description Language) from the mobile device 400 to the printer 402, e.g., over some network connection (shown via the dashed line in FIG. 4). Although XHTML-PRINT is adequate to an extent, this solution has a
20 number of drawbacks, e.g., XHTML-PRINT is not WYSIWYG, cannot control line breaks, and generally provides a poor user experience.

However, if via the present invention, the mobile device 400 first dynamically discovers a personal computer 404

equipped with a proper printer driver 406, (and which the mobile device user is authorized to use), the mobile device 400 instead provides printing data via the personal computer 404 to the printer driver 406, which in turn will create a WYSIWYG output using PCL, Postscript or rasterization. As mentioned above, the present invention provides the discovery mechanism, and thereafter the mobile device 400 may communicate in any way with the personal computer 404, such as by a lightweight PDL that has much higher-fidelity (relative to XHTML-PRINT). The result is high-quality output rendered very quickly, by leveraging the faster processor and proper printer driver 406 in the personal computer 404.

FIG. 5 demonstrates a second example, generally directed to providing an improved user experience when operating a PDA (personal digital assistant) 500 that is to be executing managed code remotely on a server 502 (e.g., an ASP.NET server). One way to accomplish this is to have the PDA connect to the server to run the managed code, (e.g., by leveraging Mobile Controls support in ASP.NET), with the results transmitted as HTML, as indicated by the dashed line in FIG. 5. The server 502 dynamically scales the HTML content to the appropriate screen size of the device 500, without the ISV creating special-purpose code for each device 500. However, this requires many round trips with the server 500,

with the screen flashing for updates, and generally provides a degraded user experience.

An improved solution provided by the present invention is accomplished by having the PDA dynamically discover that a personal computer 504 associated with the user's identity can be a proxy for a number of computational resources, as generally represented in FIG. 5. In such an event, the PDA issues an HTML request for a remote page via the proxy personal computer 504. The personal computer 504 intercepts the request and starts a session with the remote server 502 to offload the UI rendering managed code, e.g., using WSDL as the protocol to the server 502. Typically, the bandwidth from the proxy personal computer 504 to the server 502 will be less than the bandwidth from the PDA device 500 to the proxy personal computer 504, whereby the smaller amount of data transferred over the slower link to and from the server 502 results in a more responsive user interface when displayed on the PDA 500, providing an improved user experience. Of course, the two bandwidths can be first evaluated to ensure that the proxy solution is the more-optimal one.

Further, note that if the PDA 500 was equipped with locally-running controls that the server was able to leverage, an even better experience may be provided. For example, ASP.NET automatically leverages any local controls (e.g., such

as a date picker control) running locally on a PDA running a Windows[®]-based operating system. This results in even less round trips to the server 502 and/or proxy 504, with less resultant screen updating / flashing.

5 As can be appreciated, there are numerous other examples that are enabled by having computing resources revolve around an identity, instead of around the particular devices those resources ship with. The resource schemas and service fabric are not limited to simply describing the resources, but also
10 include elements to assist those resources working together. This can be appreciated from the following two types of scenarios, namely a set of real-time scenarios in which the user is interacting with a remote computing device in real-time, and a set of scheduled or predicted scenarios. An
15 example of a predicted scenario may include a storage load balancing mechanism that occasionally runs on a home network, to make sure that one computing device's free storage space is not going to run out while another drive has plenty of room.

 Thus, a rich schema may provide data that goes beyond
20 describing the resources themselves, such as methods to schedule, at a particular time or at regular intervals, prioritization machinery, and data indicating how to deal with possible contention between real-time and predicted tasks, or multiple such tasks. The schema may provide for the concept

of cost, e.g., if two connections are available to the Internet, it is valuable to recognize that the user may be paying one price for unlimited access but another price for an amount of data transferred over a different mechanism, and
5 select the better option.

Replication/sharing by a logging system may be used to understand how users are interacting with their devices, and to achieve a consolidated view of how a user operates, rather than have only disparate views separated by device. This data
10 is appropriately known to many devices, and will provide an automatic way for such data to be consolidated. Prediction optimizations may be saved and exchanged based on monitored usage, e.g., if the system knows that a user manually
downloads particular information to a hand-held computer every
15 Friday before leaving work, the system can start the download during device idle time so the "synchronization" appears instantaneous. End user options/settings/preferences is also valuable information for the shared environment, e.g., what is a desirable or even necessary feature for some users is
20 undesirable or even annoying to others. By maintaining this data, the end user has the power to be in control of their experience. Also, in a typical environment, there will be at least one rich client that uses the protocol/schema to perform system management, where "system" refers to the overall set of

resources associated with an identity. Algorithms and methods can use the schema for predicting such operations.

As is understood, there are many valuable scenarios enabled by through abstracting device resources and
5 aggregating them around an identity. A number of examples are provided, however the list should not be considered exhaustive or in any way limiting, but rather serves to illustrate some of the many benefits and advantages of the present invention. As examples mentioned above, a device with a small computer
10 screen may leverage an unused computer monitor connected to a personal computer for better display, use resources of a proxy to facilitate a better display on its own display (FIG. 5) or print to a printer even without printer driver, by connecting to a printer via a personal computer that does have the
15 correct driver (FIG. 4).

Other examples can be seen in a very low-powered device, such as a mobile telephone, performing complex computations by using the memory and CPU of a personal computer. Stored data can be automatically and transparently moved from a first hard
20 drive to a second when the first hard drive is getting full, including, for example, from personal digital video recorders that record video programming and need to delete old shows to make room for more recent recordings. The present invention instead moves those older shows to another storage, rather

than delete them. A portable audio player need not be able to play any type of physical media, but can instead receive and buffer audio streamed to it from a personal computer and play the audio from memory.

5 Other examples include allowing users to easily access their files from any device regardless of where it is stored; for example, while on the road with only a small personal computer, being able to easily open files stored on a home personal computer or work personal computer, (independent of
10 centralized .NET data storage). A user may see all of the free space in various devices, including in various public sharepoints on a corporate network. A large-screen TV connected to an appropriate device may be used to view photographs saved on a personal computer folder, run a
15 slideshow, watch home movies or movie trailers saved on a personal computer, and so on.

FIG. 7 provides a demonstration of an implementation 700 run using four interconnected computing systems, 702-705. The implementation 700 includes a command shell 710 and
20 application program 712 (or similar function code) on the computing system 704, a keyboard resource 714 on the computing system 702, a storage (e.g., disk) 716 on the computing system 703, and a display 718 on the computing system 705. Thus, in this example the resources are distributed such that there is

an input resource the on computer 702, storage resource on the computer 703, and an output resource on the computer 704. The computer 704 acts as a coordinating authority for the implementation 700. Although the resources are on different computer systems, the resources are accessed in accordance with the present invention to essentially form a single functioning system.

The shell resource 710 calls the application 712, and also connects to the other devices' resources through ports 724, 726 and 728, conceptually represented in FIG. 7 as one port per resource, although a separate input and output port may be provided. Ports essentially comprise structured queues that can be written (output) to or read (input) from via a handle to that port. Ports are mobile, data is not lost, and security can be associated with a port such that no one except the intended entity will see the data intended for it.

The shell 710 used in the implementation 700 is based on a programming language referred to as xSpresso, as described in United States provisional patent application serial number 60/379,864, filed on May 10, 2002 and herein incorporated by reference in its entirety. In general, xSpresso is directed to a programming language and runtime that brokers control and data flows among asynchronous system components. Among other things, xSpresso expresses the cooperative dimensions of

programs or services, including ordering and timing, that can
be programmatically verified. The language may
programmatically express the passing of an organizational
scheme formed from a customizable tag-based language such as
5 XML.

In the example of FIG. 7, the numbered arrows labeled 1-4
and 6-10 represent an example of the various communications
between the ports 724, 726 and 728 and the shell 710, to
display the text data of a file in the storage 716 to the
10 display 718, e.g., similar to the well-known "type" function.
In the example of FIG. 7, a user of the keyboard resource 714
initiates the operation by typing the word "type" on the
keyboard 714 (arrow labeled one (1)). The shell 710 receives
this via the port 724 and communicates data to the port 728
15 for display on the screen 718, (arrow labeled two (2)), e.g.,
to echo what the user has input and/or to prompt for a
filename of a file that stores the text that the user wants to
have displayed. In FIG. 7, the arrow labeled three (3)
represents the user typing in the filename, such as
20 corresponding to an XML file, which may also be echoed (not
separately labeled).

With the filename, the shell 710 then writes data
(represented by arrow labeled four (4)) to the port 726 that
corresponds to the storage (e.g., disk) resource 716, which

results in a read request to the storage's file system. In parallel, the shell 710 calls (the wide arrow labeled five (5)) the application program 712 to receive and output the read data. In response, the storage resource 716 begins
5 outputting data via port 726 to the application program 712, which in turn outputs it to the port 728 of the display resource 718, as represented by the arrows labeled six (6) and seven (7). The arrows labeled eight (8) and nine (9) represent the rest of the data being transferred in a similar
10 manner. The arrow labeled ten (10) represents the status (e.g., success) code being returned to the shell 710.

In keeping with the present invention, storage, input and output can be placed on different systems, with no change to the implementation. Legacy interoperability may be provided
15 via common language runtime (CLR) classes that represent common legacy components (e.g., the keyboard, console and storage). System dispersion is provided in a local and transparently distributed implementation, as is explicit handling of concurrency and message-passing.

20 As can be seen from the foregoing detailed description, there is provided a method and system for distributed computing of device resources based around the concept of identity. The method and system abstracts the resources of a computer from the individual devices in a manner that provides

numerous benefits to a user and generally improves the user's computing experience. The method and system are schema-based, and are extensible and highly flexible.

While the invention is susceptible to various
5 modifications and alternative constructions, certain
illustrated embodiments thereof are shown in the drawings and
have been described above in detail. It should be understood,
however, that there is no intention to limit the invention to
the specific forms disclosed, but on the contrary, the
10 intention is to cover all modifications, alternative
constructions, and equivalents falling within the spirit and
scope of the invention.

WHAT IS CLAIMED IS:

1. In a computing environment, a method comprising:
at a first computing device, discovering a second
computing device;

5 at the first device, obtaining data related to the second
device, the data including resource information of the second
device formatted according to a schema understood by the first
device; and
the first device interpreting the data to access a
10 resource of the second device.

2. The method of claim 1 further comprising,
communicating a request to use the resource from the first
device to the second device.

15

3. The method of claim 1 further comprising,
communicating identity information from the first device to
the second device.

20

4. The method of claim 3 wherein the identity
information corresponds to credentials.

5. The method of claim 4 wherein the credentials are
provided by an authentication service.

6. A computer-readable medium having computer-executable instructions to perform a method, the method comprising:

5 at a first computing device:

locating information about a resource of at least one other device that is available for use, the information located based on an identity associated with the first computing device;

10 selecting a resource of a second device from the information; and

accessing the resource of the second device.

7. In a computing environment, a system comprising:

15 a first computing device associated with at least one identity, the first device including a first resource and configured to provide access to the first resource by at least one other device associated with the identity, the first computing device providing information about the first
20 resource according to a resource schema associated with the at least one identity;

a second device, the second device configured to request access to the first resource; and

the second device performing a computing task including using the first resource on the first device to perform at least part of the task.

5 8. The system of claim 7 wherein the first resource comprises storage.

9. The system of claim 7 wherein the first resource comprises bandwidth of a connection from the first device to
10 at least one other computing device.

10. The system of claim 7 wherein the first resource comprises processing cycles of a processor.

15 11. The system of claim 7 wherein the first resource comprises an input method.

12. The system of claim 7 wherein the first resource comprises an output method.

20

13. The system of claim 7 further comprising a third computing device connected to the first computing device, wherein the second device accesses the first resource to perform an operation with the third computing device.

14. The system of claim 7 wherein the third computing device comprises a printer, and wherein the first resource corresponds to a printer driver on the first computing device.

5

15. The system of claim 7 wherein the third computing device comprises a server, and wherein the first computing device acts as a proxy to the server for the second computing device.

10

16. The system of claim 15 wherein the bandwidth from the second computing device and the first computing device is greater than the bandwidth from the first computing device to the third computing device.

15

17. A computer-readable medium having stored thereon a data structure, the data structure comprising:

a first field for identifying a type of resource;

a second field for identifying a device that includes a resource corresponding to the type of resource identified in the first field;

20

a set of at least one third field, each third field in the set corresponding to the second field for identifying

information about the device resource identified by the first field; and

wherein the data structure is used to provide resource information to another device that is different from the one
5 identified in the second field such that the other device can use a resource identified in the third field set.

18. The data structure of claim 17 wherein the first field includes information corresponding to a storage resource
10 type.

19. The data structure of claim 18 wherein the third field set includes information corresponding to total size data of the storage resource.
15

20. The data structure of claim 18 wherein the third field set includes information corresponding to free space data of the storage resource.

20 21. The data structure of claim 17 wherein the first field includes information corresponding to bandwidth data.

22. The data structure of claim 17 wherein the wherein the first field corresponds to processing power.

23. The data structure of claim 22 wherein the third field set includes information corresponding to speed data of a processor.

5

24. The data structure of claim 22 wherein the third field set includes information corresponding to usage data of a processor.

10

25. The data structure of claim 22 wherein the third field set includes information corresponding to size data of a memory.

15

26. The data structure of claim 25 wherein the third field set includes information corresponding to usage data of a memory.

20

27. The data structure of claim 17 wherein the second field includes information identifying the device as a personal computer.

28. The data structure of claim 17 wherein the second field includes information identifying the device as a disk drive.

791

29. The data structure of claim 17 wherein the second field includes information identifying the device as a telephone.

5

465

ABSTRACT

A system, method and related data structures for
discovering and describing computing resources available at
various computing devices, and for exposing those resources as
5 services that are addressable by software applications. The
data describing the resources is arranged according to an
identity-based schema. The computing resources may include,
for example, storage capacity, bandwidth, processing power,
input methods and mechanism, and rendering methods. The
10 method and system are identity-based, whereby a user (with an
identity) has access to the distributed resources commensurate
with that identity.

793

EPO - Munich
55
30. April 2003

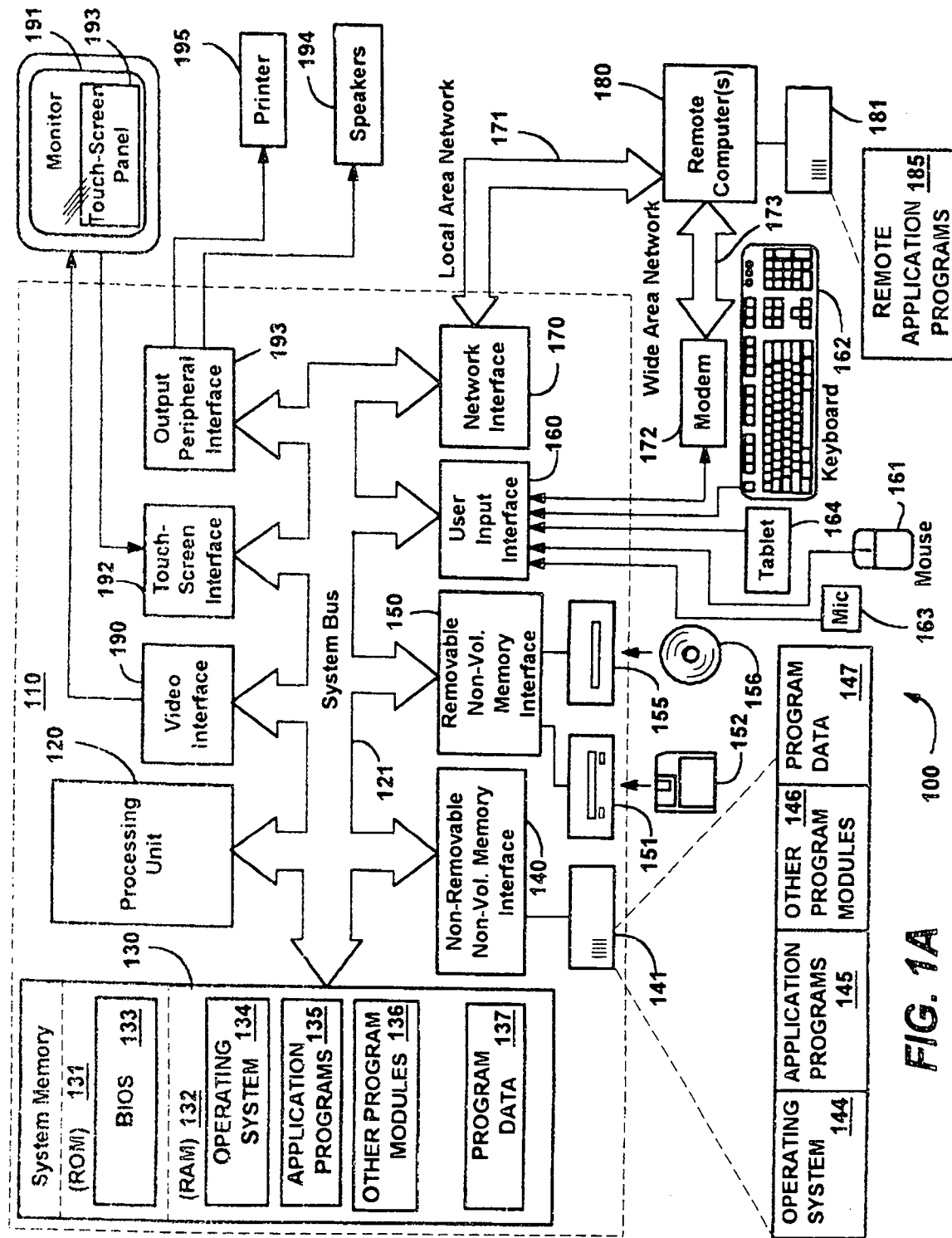


FIG. 1A

467

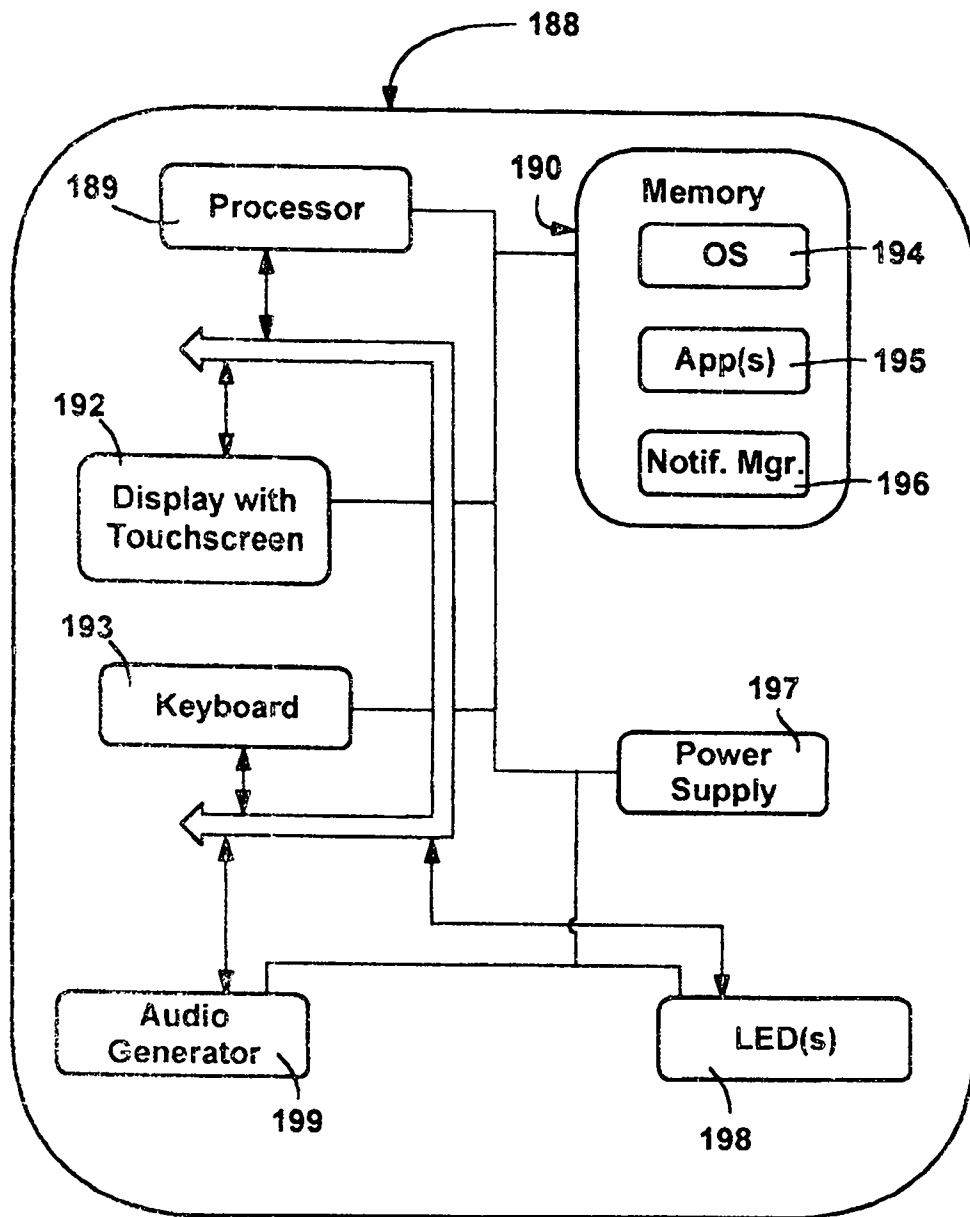


FIG. 1B

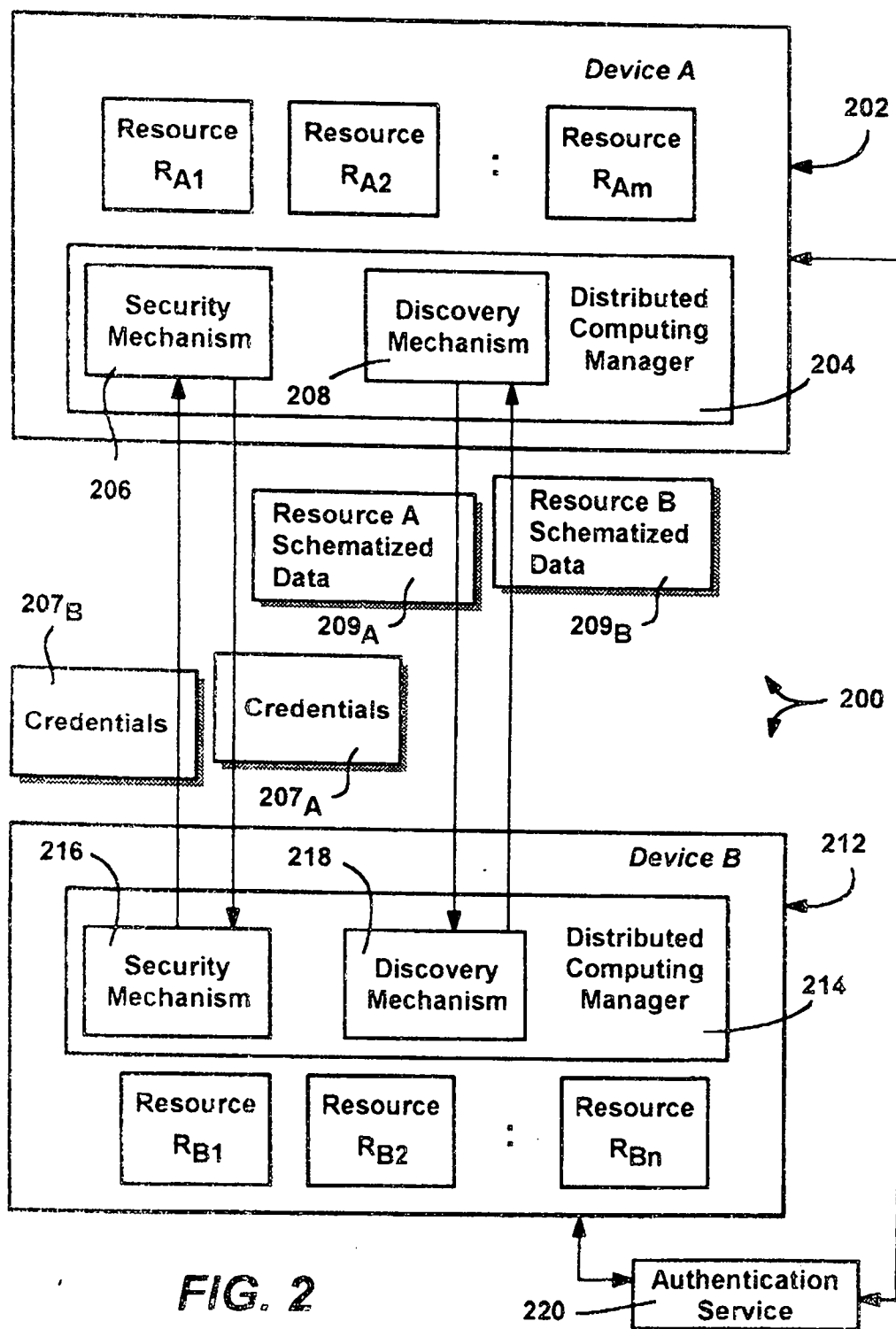


FIG. 2

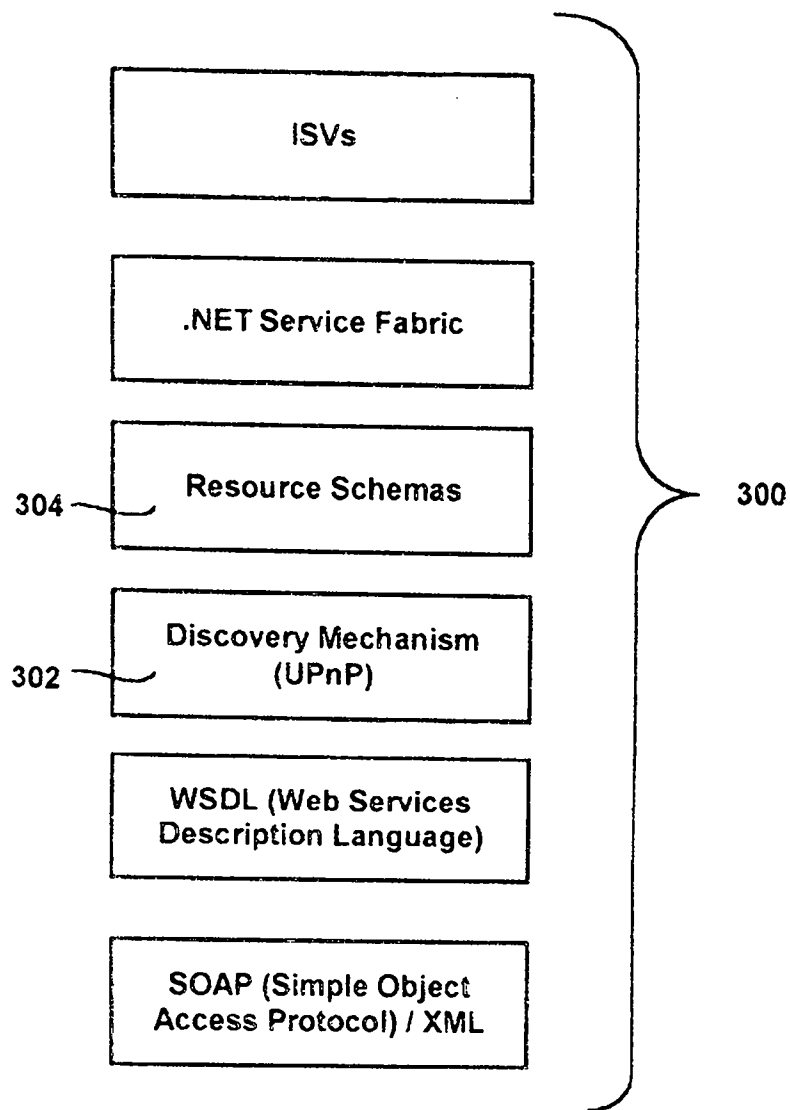


FIG. 3

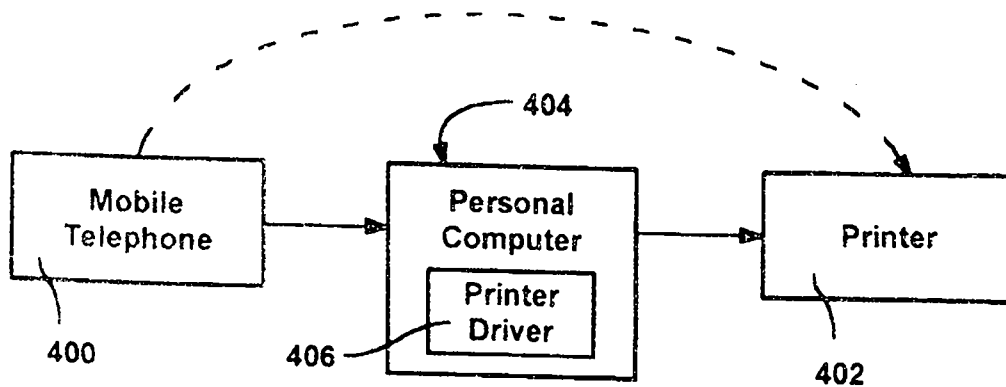


FIG. 4

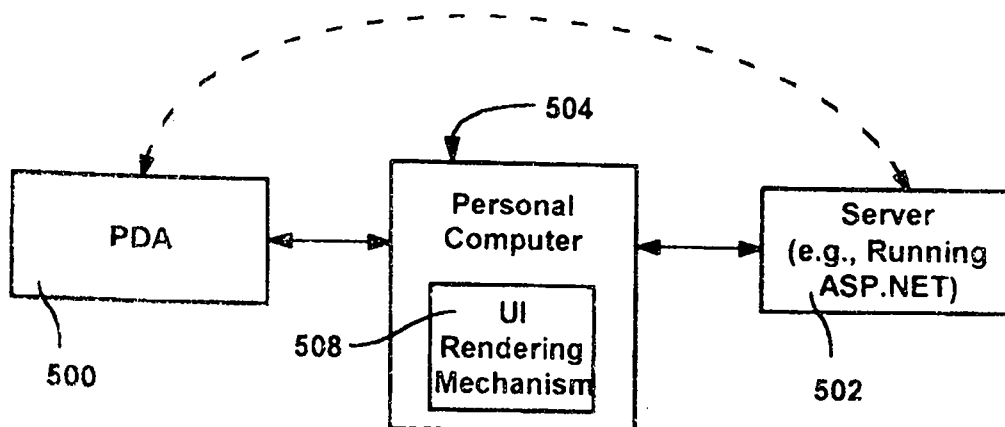
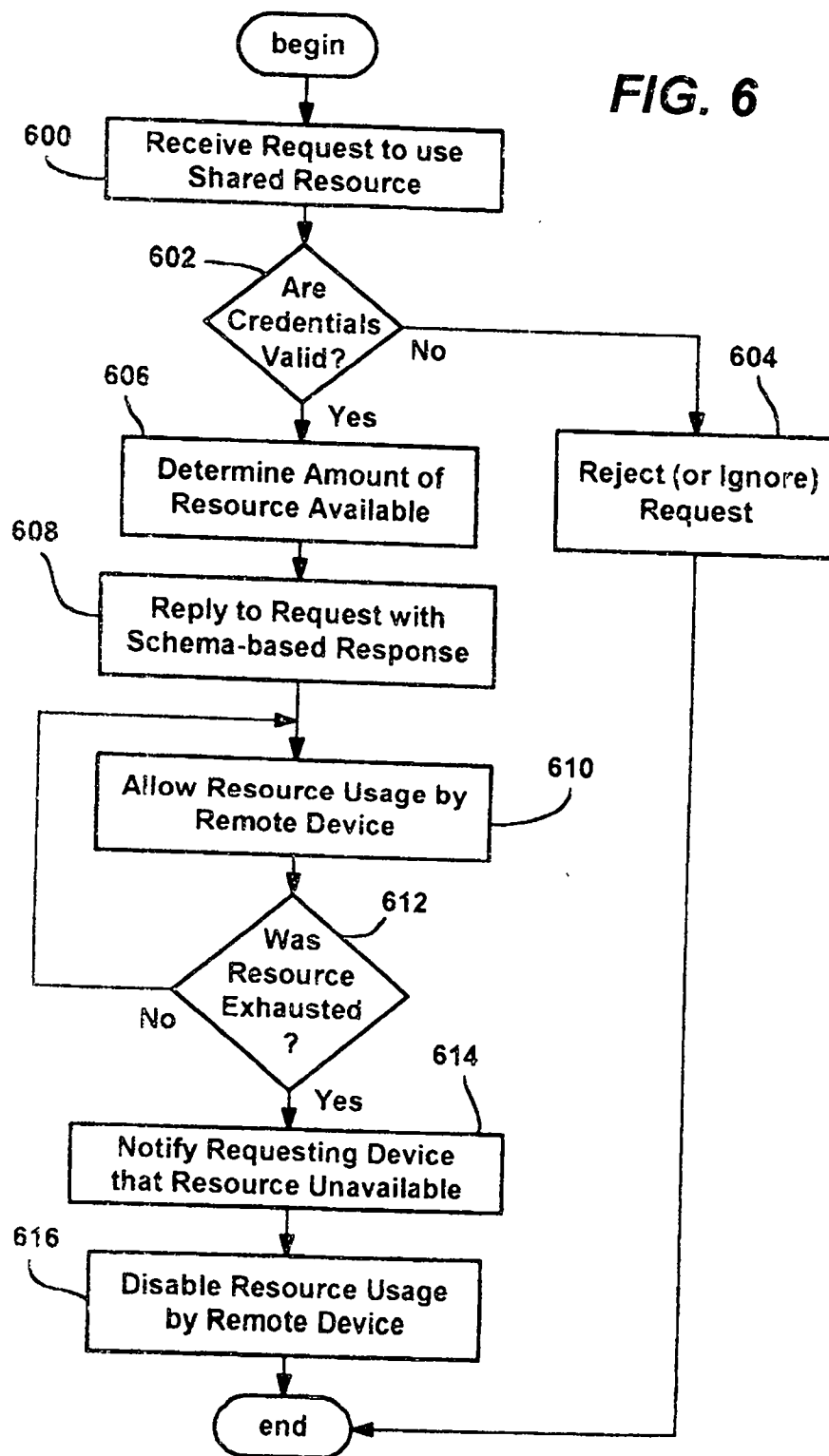


FIG. 5

FIG. 6



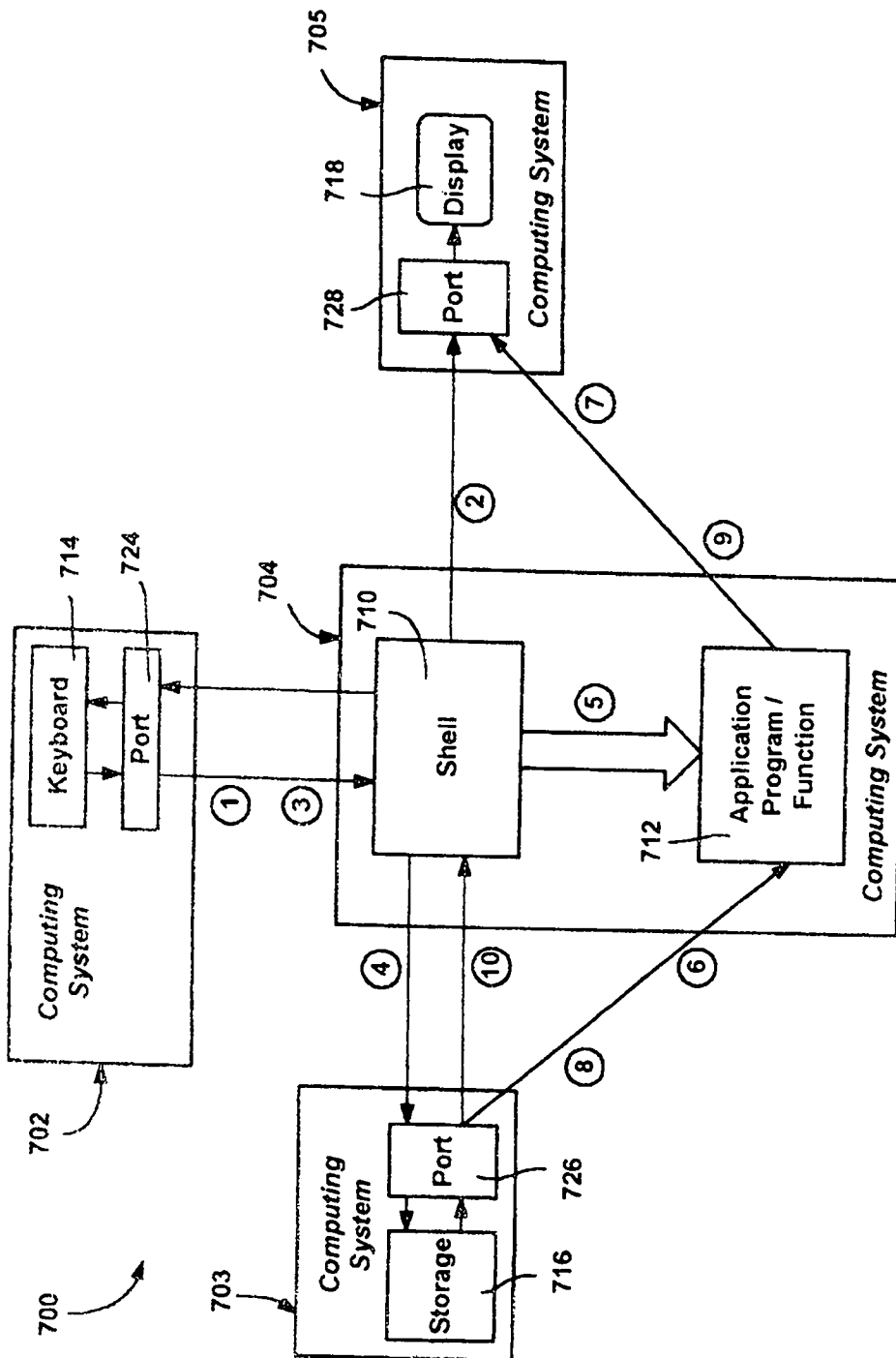


FIG. 7

799

423

Bogotá D.C., febrero 7 de 2008

Doctora
Diego González
Deloitte & Touche Ltda.
Carrera 7 # 74-09 Piso 7
Ciudad

En nuestra calidad de Asesores Legales de la sociedad Ogilvy & Mather S.A. informamos a ustedes que a la fecha no tenemos conocimiento de ningún juicio o litigio de importancia que pudiera afectar sustancialmente a la sociedad mencionada.

Tampoco tenemos conocimiento de ningún reclamo de impuestos, o de cualquier otro pasivo o contingente que afectara el patrimonio de la sociedad a la fecha indicada.

Así mismo, les informamos que al 31 de diciembre de 2007 existía un saldo pendiente de pago por concepto de honorarios de S ____.

Atentamente,

CARMEN SANABRIA GOMEZ

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION- 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

LA SUPERINTENDENTE DE INDUSTRIA Y COMERCIO (E)

en ejercicio de sus facultades legales, en especial de las que se confirieron en el artículo 4, numeral 5 del decreto 2153 de 1992, y

CONSIDERANDO

PRIMERO: Que mediante escrito radicado en esta Superintendencia el 4 de junio de 2003, bajo el número 03046838, se presentó la solicitud de patente de invención titulada "COMPUTACIÓN DISTRIBUIDA BASADA EN IDENTIDAD PARA RECURSOS DE ARTEFACTOS".

SEGUNDO: Que efectuado el examen de forma en cuanto a los requisitos que debe llenar la solicitud y cumplidas las exigencias legales, se ordenó publicar el extracto sin que se hubieran presentado oposiciones por parte de terceros.

TERCERO: Que realizado el examen de patentabilidad se encontró que la invención no es patentable. En consecuencia, con fundamento en lo dispuesto en el artículo 45 de la Decisión 486 de la Comisión de la Comunidad Andina, se requirió al peticionario para que se pronunciara sobre las objeciones efectuadas a la solicitud.

CUARTO: Que mediante escrito radicado el 19 de julio de 2006, el solicitante atendió oportunamente el requerimiento formulado y presentó aclaraciones. Así mismo, allegó un nuevo capítulo reivindicatorio, el cual se ajusta a las prescripciones del artículo 34 de la Decisión 486.

QUINTO: Que según lo dispuesto en el artículo 14 de la Decisión 486 expedida por la Comisión de la Comunidad Andina se otorgarán patentes "...para las invenciones, sean de producto o de procedimiento en todos los campos de la tecnología, siempre que sean nuevas, tengan nivel inventivo y sean susceptibles de aplicación industrial".

SEXTO: Que las nuevas reivindicaciones 1 a 29, contenidas en los folios 279 a 283, además de no cumplir con el requisito de claridad, no se considera una invención y pretenden proteger un uso distinto, de acuerdo con las siguientes consideraciones:

1. Objeto de la invención

Según la descripción, el objeto de la solicitud consiste en suministrar un sistema y un método para descubrir y describir los recursos de computación disponibles en varios dispositivos de computación asociados con una identidad y exponer esos recursos como

El Suscrito Secretario General, AD - HOG

Hace Constatar que la presente **Sede Centro:** Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Sede CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Teléfono: 382 26 95. Línea 9800-910 165
Fax: 382 26 95. Línea 9800-910 165
Web: www.sic.gov.co **e-mail:** info@sic.gov.co
Bogotá D.C. - Colombia

tenido a la vista

Secretaría General AD - HOG

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

servicios a los que se puede acceder con programas de software tal que los programas de los dispositivos apalquen los recursos de otro dispositivo.

Por su parte, el nuevo capítulo reivindica en un ambiente de computación, un método, relacionado en las reivindicaciones 1 a 5; un medio que se puede leer por computador que tiene instrucciones ejecutables por computador para realizar un método, relacionado en la reivindicación 6; en un ambiente de computación, un sistema, relacionado en las reivindicaciones 7 a 16; un medio que se puede leer por computador que tiene almacenado en el mismo una estructura de datos, una estructura, relacionado en la reivindicación 17; y, una estructura de datos, relacionada en las reivindicaciones 18 a 29.

2. Sobre el nuevo capítulo reivindicatorio

2.1 Reivindicaciones

El artículo 30 de la citada Decisión 486 establece que "las reivindicaciones definirán la materia que se desea proteger mediante la patente. Deben ser claras y concisas y estar enteramente sustentadas por la descripción".

Las reivindicaciones 1, 6, 7, 17 y sus dependientes son indefinidas, porque no permiten establecer cual es la contribución técnica al arte.

Las reivindicaciones 1, 6, 7 y 17 son muy generales, porque se relacionan con un método, un medio legible por computador y un sistema cualquiera; además, junto con sus dependientes, presentan como pasos de un método, como medios de un sistema una secuencia de procesamiento de datos ejecutada por computador.

Dado lo anterior, se advierte que las mencionadas reivindicaciones no cumplen con el requisito de claridad, no obstante fueron evaluadas y calificadas.

2.2. Reivindicaciones de producto o de procedimiento

El artículo 14 de la Decisión 486 establece claramente que los países miembros otorgarán patentes para las invenciones, sean de producto o de procedimiento.

Para dar cumplimiento a tal exigencia, y como la misma normativa andina no establece la forma como se deben definir las reivindicaciones de producto y las de proceso, la Oficina, en la práctica común, ha acudido a las indicaciones de la 'Guía para los países en desarrollo sobre el examen de las solicitudes de patentes', la cual fue publicada por la OMPI en 1982 con el propósito de ayudar a los examinadores de la oficinas nacionales en sus estudios.

 Precisamente, en el párrafo 57 (de dicha guía) se especifica que existen dos

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO
Sede Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Sede CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Fax: 382 26 95. Línea 9800-910 165
Hace Constar que la presente copia fotostática coincide con el original que ha sido tenido a la vista
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341
Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

tipos de reivindicaciones: de *producto* y de *procedimiento* (método). Adicionalmente, el citado párrafo indica que las reivindicaciones de *producto* deben definir, el objeto que se desea proteger, mediante las características estructurales ó de función.

De allí, es evidente que un *producto*, desde el punto de vista técnico, debe ser definido o bien por las características de estructura o bien por las características de funcionamiento entre los componentes de una estructura pero no mediante mezcla de ellas.

En el presente caso, las reivindicaciones del pliego calificado no definen un producto o un procedimiento (método) de acuerdo a la práctica establecida, sino que se relacionan, de forma explícita, a un soporte lógico como tal.

3. No se considera invención

El artículo 15e) de la Decisión 486 señala, que no se considerarán invenciones: "los programas de ordenadores o el soporte lógico como tales".

Al estudiar la solicitud, es evidente que tanto la descripción como las reivindicaciones, que se refieren a: "**en un ambiente de computación un método**", "un medio que se puede leer por computador que tiene **instrucciones ejecutables por computador**", "**en un ambiente de computación un sistema**", "un medio que se puede leer por computador que **tiene almacenado en el mismo una estructura de datos**", relacionan unas secuencias lógicas de procesamiento de datos (información), llevadas a cabo mediante el uso de medios técnicos como, por ejemplo, el ambiente informático, de implementación convencional para exponer recursos como servicios a los que se accede con programas de software, el cual, se ilustra en las figuras 1A y 1B junto con la descripción correspondiente.

Dichos medios permiten procesar la información (las señales) del modo deseado, por ejemplo, para lograr el resultado relacionado en las reivindicaciones; por esta razón, el pliego es un soporte lógico como tal, que no se considera una invención de conformidad con lo dispuesto en el citado artículo 15e).

Por otra parte, se desconoce, además, lo dispuesto en el artículo 21 de la Decisión 486, porque se hace otro (*segundo*) uso de dispositivos convencionales o de partes de ellos, que en este caso consiste en el uso de un ambiente de cómputo, además, porque de forma explícita se describe y reivindica que el objeto se realiza en un ambiente de computación.

Por las anteriores razones, se considera que lo reivindicado en el presente caso, se encuentra incurso en lo dispuesto por los artículos 15 inciso e) y 21 de la norma supranacional andina.

El Suscrito Secretario General, AD - IIC

Hace Constar que la presente resolución es legal y válida y que la prescripción formal coincide con el ordenamiento jurídico colombiano.
Sede Gan: Tr. 40A, No. 38-50 Conmutador: 3820840
Fax: 382 26 95. Línea 9800-910 165

tenido a la vista

Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia

Administración General de la Industria y el Comercio

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

4. Respuesta del solicitante

Sobre los comentarios del solicitante a la notificación de fondo, conviene realizar las siguientes consideraciones:

- Con referencia a los puntos 1 a 3 de dicha respuesta, se puntualiza que el nuevo pliego se aceptó y calificó acorde a las disposiciones vigentes; sin embargo, es necesario precisar que las correcciones de las reivindicaciones se limitaron a la introducción de la frase "caracterizado por" en lugar de los términos objetados y se mejoró la redacción al cambiar el término "artefacto" por "dispositivo"; sin embargo, el contenido sigue siendo el mismo que se cuestionó en el Oficio n° 4276 notificado al solicitante el 20 de abril de 2006 (de aquí en adelante llamado oficio) y, así lo señala el mismo interesado cuando afirma, "...modificando levemente el capítulo...".

Por las afirmaciones que se hacen en la respuesta, es evidente que supone que por haber introducido la frase "caracterizado por" en las reivindicaciones, estas quedan completamente definidas y, por ello, que cumplen los requisitos del artículo 30, lo cual, no es cierto, por las razones que a continuación se exponen.

De un lado, en el oficio se dijo, y aquí se reitera, que las reivindicaciones 1, 6, 7, 17 son indefinidas, porque no es posible establecer cual es el aporte del inventor al estado de la técnica. Al respecto, se hace necesario aclarar que la esencia de las reivindicaciones consiste en definir la invención, con lo cual, se fija la extensión de la patente y se hace posible delimitarla respecto del estado de la técnica; además una reivindicación independiente debe definir la invención mediante características técnicas. Asimismo, es práctica corriente subdividir las reivindicaciones en dos partes, preámbulo y parte caracterizante; la primera, contiene el conjunto de características de la invención cuya combinación se conoce, por decirlo así, es el "género" al que pertenece la invención, y la segunda parte (caracterizante) debe mencionar la(s) característica(s) propia(s) de la invención (creadas por el inventor) que la distinguen del estado de la técnica, por decirlo así, es la "especie" de la invención. Por último, las dependientes son para aclarar o delimitar alguna característica incluida en una independiente.

De otro lado, para que un pliego sea definido debe estar acorde con lo señalado por el artículo 14, es decir, referirse a un *producto* o un *procedimiento*, lo cual, no sucede en este caso, porque, "en un ambiente de computación, un método", "un medio que se puede leer por computador que tiene instrucciones ejecutables por computador para realizar un método", "en un ambiente de computación, un sistema" y "un medio que se puede leer por computador que tiene almacenado en el mismo una estructura de datos, una estructura", no son un *procedimiento* ni un *producto* definidos en los términos establecidos.

**SUPERINTENDENCIA DE INDUSTRIA Y
COMERCIO**

El Suscrito Secretario General
Hace Constar que la presente
tática coincide con el original que se
tenido a la vista

Sede Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Sede CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Fax: 38212695. Línea 9800-910 165
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

A la luz de lo anterior, las reivindicaciones 1, 6, 7, 17 son indefinidas no solo por no referirse a un *procedimiento* sino porque no hay una clara separación entre "género" y "especie", además, las características allí relacionadas son una secuencia de procesamiento de datos realizados con computador, lo que reiteramos es *soporte lógico como tal*.

Por ejemplo, en la respuesta se asegura que, "... las reivindicaciones están redactadas de manera adecuada...", lo cual, no demuestra que las reivindicaciones no fueron redactadas como secuencias de procesamiento de datos y que estén acorde con lo dispuesto en los artículos 14 y 30. Es obvio que, hacer aseveraciones sin sustento alguno, no es fundamento para controvertir las objeciones planteadas.

- Para controvertir los comentarios del punto 4 de la respuesta, cabe señalar que el objeto de la solicitud no se define acorde a la práctica establecida por la oficina, basada en las definiciones del párrafo 57 de la *Guía sobre el examen de las solicitudes de patentes* y en especial la jurisprudencia dictada al respecto.

En efecto, tradicionalmente el Tribunal de Justicia de la Comunidad Andina se ha pronunciado sobre el tema de los *productos* y de los *procedimientos*. Concretamente, dentro del proceso 36-IP-98, fijó unas pautas acerca de lo que es *producto* y *procedimiento*, como se aprecia en punto 4.1, del citado proceso que establece:

"(...) Las invenciones de producto son aquellas (...) que revisten una forma tangible, por ejemplo, máquinas, equipos, aparatos, dispositivos, etc. la invención puede residir tanto en un producto independiente como en un producto que solo constituye una parte de otro producto y que solo puede venderse como parte de otro producto. Una invención de procedimiento, en general, es una solución consistente en una secuencia de etapas conducentes a la fabricación de un producto (...)", (subrayados fuera de texto).

A la luz de lo anterior, las aseveraciones que se hacen en la respuesta dejan sin sustento alguno a las reivindicaciones del pliego, puesto que, de un lado, el *procedimiento* (método) reivindicado no consiste en una secuencia de etapas conducentes a la fabricación de un producto sino que relacionan una secuencia de procesamiento de información con medios técnicos y las de *producto* no revisten una forma tangible, porque en ellas se hace referencia a elementos lógicos de programación que se usan en medios técnicos de manera que no existe una forma tangible de realización, motivo por el cual las reivindicaciones quedan sin sustento real.

En conclusión, hacer una serie de aseveraciones sin sustento, no sirve de fundamento para controvertir las objeciones planteadas en el oficio trasladado.

**SUPERINTENDENCIA DE INDUSTRIA Y
COMERCIO**

El Suscrito Secretario General,
Hace Constatar que la presente copia
tática coincide con el original
tenido a la vista

Sede Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Sede CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Fax: 382 26 95. Línea 9800-910 165
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia

Secretario General AD-HOC

321

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

- Para desvirtuar los comentarios del punto 5 de la respuesta, es importante señalar que el *procedimiento* (métodos) y los *productos* reivindicados no consisten ni en una secuencia de etapas conducentes a la fabricación de un producto ni revisten una forma tangible, como se dijo antes, sino que, contrario a lo que argumenta el interesado, se refieren a secuencias de procesamiento de datos realizadas por computador, que en síntesis es el USO de medios técnicos para el procesamiento de datos (información) no técnicos.

Aún más, con la sola lectura de la especificación se puede comprobar, fácilmente, que las reivindicaciones no definen ni un *procedimiento* (método) ni un *producto*, sino que se relacionan, de forma explícita, a un soporte lógico como tal.

Para constatar lo anterior conviene, por ejemplo, dirigirse a los folios 10 a 23 de la descripción, donde se dice: "la figura 1A ilustra un ejemplo de un ambiente de sistema de computación adecuado para implementar el invento... alcance de la utilización (uso)...", en la página 11, se afirma, "... el invento se puede describir en el contexto general de instrucciones ejecutables por computador, tales como módulos de programas, que son ejecutados por un computador...".

Adicionalmente, se reitera, las aseveraciones formuladas en la respuesta son infundadas. Al respecto, cabe anotar que con solo afirmar que en la solicitud se define un *procedimiento* o un *producto* o calificar de 'irrelevante' la argumentación del examinador, no son argumentos que demuestren que lo descrito y reivindicado no sea un soporte lógico (software) como tal.

Aún más, no queda claro en que se fundamenta el interesado para afirmar que, "... el resultado final logrado... se implementa en un computador o cumple con fines de programación, este hecho no es relevante para desvirtuar la patentabilidad de los objetos... resulta improcedente considerar las reivindicaciones como un soporte lógico...", ni explica por qué se debería entender así.

En conclusión, contrario a las suposiciones del interesado, en ningún lugar de su respuesta se demuestra que lo solicitado no es soporte lógico como tal.

- Con relación a los comentarios del punto 6 de la respuesta, este Despacho puntualiza que los párrafos que el interesado dice no pudo entender, tuvieron el fin de aclararle la posición de la Oficina y de hacerle entender al interesado que la solicitud no contiene elementos que puedan servir de base para reclamar ni un *procedimiento* ni un *producto*, definidos acorde a lo que establecen la norma y la práctica administrativa vigentes, además que no se cumplen ciertos criterios de patentabilidad claramente especificados en el oficio.

SUPERINTENDENCIA DE INDUSTRIA Y

COMERCIO

Sede-Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10

Sede-CAN: Tr. 40A No. 38-50 Conmutador: 3820840

Cófax: 382 26 95. Línea 9800-910 165

El Suscrito Secretario Hace Constatar que la práctica coincide con el original que se tiene a la vista

Web: www.sic.gov.co e-mail: info@sic.gov.co

Bogotá D.C. - Colombia

Director General AD UCO

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

Rad. Exp. N° 03046838

Es evidente que el Oficio trasladado cuestiona la claridad, el soporte lógico, los segundos usos y la falta de nivel inventivo, es decir, la patentabilidad de la solicitud, y entre los requisitos de aquella, está el de nivel inventivo.

- Sobre los comentarios de los puntos 7 y 8 de la respuesta, son una larga iteración de lo ya debatido antes y que, igualmente, se reitera, no tienen sustento alguno por las siguientes razones:

a) el procedimiento (método) reivindicado no consiste en una secuencia de etapas conducentes a la fabricación de un producto sino que relacionan una secuencia de procesamiento de información con medios técnicos; de forma similar, los productos no revisten una forma tangible por las razones arriba explicadas; b) las reivindicaciones relacionan unas secuencias lógicas de procesamiento de datos (información), llevadas a cabo mediante el uso de medios técnicos de implementación convencional o ampliamente conocida en la técnica; c) la divulgación se refiere a un soporte lógico como tal, como se puede comprobar con ayuda de los folios 10 a 23 de la descripción; d) los argumentos trasladados en el oficio, el interesado los pretende descalificar con meras afirmaciones que no tienen sustento técnico ni jurídico; y, e) si bien las invenciones pueden emplear materiales conocidos, su patentabilidad siempre debe estar dada por la combinación creativa y sorprendente de tales materiales conocidos, de manera que implique un avance real en el estado de la técnica.

En vista de lo expuesto, se puede concluir que la solicitud no cumple lo dispuesto en los artículos 14, 15e), 21 y 30, y teniendo en cuenta que los comentarios y correcciones allegados no controvierten ni subsanan en forma alguna las objeciones planteadas en el oficio N° 4.276, se considera que la solicitud no puede acceder al privilegio solicitado.

RESUELVE

ARTICULO PRIMERO: Negar el privilegio de patente de invención a la creación denominada: "COMPUTACIÓN DISTRIBUIDA BASADA EN IDENTIDAD PARA RECURSOS DE ARTEFACTOS".

ARTICULO SEGUNDO: Notificar personalmente al doctor Darío Cárdenas Navas, en su calidad de apoderado de la sociedad Microsoft Corporation, o a quien haga sus veces, el contenido de la presente resolución, entregándole copia de la misma, advirtiéndole que

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO
COMERCIO
El Suscrito Secretario General, AD - HCC
Hace Constar que la presente resolución coincide con el original
tenido a la vista
Sede: Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Sede: CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Fax: 382 26 95. Línea 9800-910 165
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia
Secretario General AD - HCC

3223

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

RESOLUCION 11341

Por la cual se niega una patente de invención

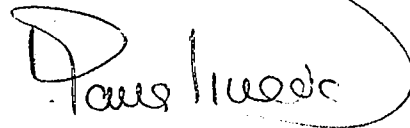
Rad. Exp. N° 03046838

contra ella procede el recurso de reposición ante el Superintendente de Industria y Comercio al momento de notificarse o dentro de los 5 días hábiles siguientes.

NOTIFIQUESE Y CUMPLASE

Dada en Bogotá D.C., a los 17 ABR 2004

LA SUPERINTENDENTE DE INDUSTRIA Y COMERCIO (E),



MARIA TERESA PINEDA BUENAVENTURA

Doctor
DARIO CARDENAS NAVAS
Carrera 7 N° 71 – 52
Torre B Piso 9°
Bogotá D.C.

**SUPERINTENDENCIA DE INDUSTRIA Y
COMERCIO**

El Suscrito Secretario General
Hace Constar que la presente
copia coincide con el original
entido a la vista

Sede Centro: Carrera 13, No. 27-00 Pisos 2, 5, 7 y 10
Sede CAN: Tr. 40A No. 38-50 Conmutador: 3820840
Fax: 382 26 95, Línea 9800-910 165
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia

Secretario General AD-1000