

EJECUCIÓN EXTERNALIZADA DEL EDITOR DE MÉTODO DE ENTRADA

Antecedentes de la invención

5 Un editor de método de entrada, o "IME", es un programa que transforma los eventos de entrada del usuario en texto especificado por parte del usuario para el consumo por medio de una aplicación, como para ampliar un documento editado por la aplicación. Por ejemplo, un teclado en la pantalla, que
10 realiza la escritura mediante trazos, completación automática de palabras, predicción de palabras o composición de caracteres ideográficos, puede implementarse como un IME, al igual que un mecanismo de transcripción automática de voz.

 Cuando un usuario selecciona un IME particular para su
15 uso con una aplicación, el sistema operativo normalmente provoca que el IME se cargue en el proceso de la aplicación y se ejecute en el mismo. El IME almacena el texto que genera en el documento dentro del espacio de dirección del proceso de la aplicación. Se utiliza un bloqueo u otro mecanismo de
20 sincronización para evitar que la aplicación y el IME escriban en el documento al mismo tiempo, lo cual podría dañar el documento.

Breve descripción de la invención

25 Esta breve descripción se proporciona para introducir una

selección de conceptos en una forma simplificada que, además, se describe más adelante en la descripción detallada. Esta breve descripción no tiene por objeto identificar los factores fundamentales o las características esenciales de la materia objeto reivindicada ni tampoco pretende limitar el alcance de la materia objeto reivindicada.

Se describe un recurso para procesar la entrada textual generada con un dispositivo de entrada de usuario. En un recurso ejemplar, el recurso almacena la entrada textual en una primera estructura de datos contenida en el espacio de dirección de un primer proceso. El recurso sincroniza el contenido de la primera estructura de datos con una segunda estructura de datos contenida en el espacio de dirección de un segundo proceso distinto del primer proceso, una aplicación que se ejecuta en el segundo proceso se configura para consumir una entrada textual sincronizada con la segunda estructura de datos.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques que muestra algunos de los componentes que pueden incorporarse en al menos algunos de los sistemas informáticos y otros dispositivos en los cuales opera el recurso.

La figura 2 es un diagrama de flujo de datos que representa la operación del recurso en algunos ejemplos.

La figura 3 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del IME recibidas por el servidor de entrada de texto.

5 La figura 4 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del cliente de entrada de texto recibidas por el servidor de entrada de texto.

La figura 5 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para gestionar el tamaño de la memoria tampón de edición.

La figura 6 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del servidor de entrada de texto recibidas por el cliente de entrada de texto.

La figura 7 es un diagrama de comunicación que muestra un ejemplo de las interacciones entre el servidor de entrada de texto del recurso y su cliente de entrada de texto.

20

Descripción detallada de la invención

Los inventores han identificado las desventajas significativas con las metodologías convencionales para ejecutar los IME. En particular, los inventores reconocen que el proceso de aplicación está sobrecargado por su ejecución del

25

IME, tanto por el consumo de los recursos del proceso de solicitud del IME como por el tiempo que la aplicación pasa bloqueada durante el bloqueo del documento.

Además, los inventores han determinado que constituye un riesgo de seguridad permitir que el código para la aplicación y el código para el IME se ejecuten en el mismo proceso, lo cual puede permitir que un fallo en el IME bloquee la aplicación.

Además, los inventores han reconocido que la ejecución de los IME en el mismo proceso que la solicitud afecta la capacidad de la aplicación de ejecutarse en un sistema informático distinto del sistema informático al cual están directamente conectados los dispositivos de entrada de usuario.

Con el fin de superar estas desventajas, los inventores han concebido y se han limitado a la práctica de un recurso de *software* y/o *hardware* para ejecutar un editor de método de entrada externo al proceso de la aplicación que recibe su entrada ("el recurso").

En algunos ejemplos, el IME se ejecuta en un proceso de servicio de entrada — ya sea en el mismo sistema informático que la aplicación o en un sistema informático diferente — y almacena el texto que genera en un proxy para el documento de la aplicación en el proceso de servicio de entrada denominado memoria tampón de edición. Un servidor de entrada de texto que se ejecuta en el proceso de servicio de entrada y un cliente de entrada de texto que se ejecuta en el proceso de aplicación

se comunican para mantener la memoria tampón de edición sincronizada con un control de edición a través del cual la aplicación recibe la entrada generada por el IME.

En algunos ejemplos, el recurso actualiza la memoria
5 tampón de edición cada vez que el texto se agrega, elimina o modifica por el IME, y comúnmente cuando el texto se agrega, elimina o modifica por el control de edición o la aplicación.

En algunos ejemplos, para restringir la cantidad de almacenamiento ocupada por la memoria tampón de edición y la
10 cantidad de comunicación entre procesos que se produce entre el proceso de servicio de entrada y el proceso de aplicación, el recurso mantiene el contenido incompleto del control de edición en la memoria tampón de edición. Para ello, el cliente de entrada de texto mantiene un seguidor de memoria caché en la
15 memoria tampón en el proceso de aplicación para realizar el seguimiento de qué porciones del texto en el control de edición se almacenan en memoria caché en la memoria tampón de edición. Cuando el IME cambia el punto de inserción o el intervalo de selección de texto, el servidor de entrada de texto
20 notifica al cliente de entrada de texto y el cliente de entrada de texto comprueba el seguidor de memoria caché en la memoria tampón para determinar si una región que rodea el nuevo punto de inserción/selección se almacena en memoria caché por la memoria tampón; si no es así, el cliente de entrada de texto
25 envía esta porción al servidor de entrada de texto para el

almacenamiento en la memoria tampón de edición y actualiza el seguidor de memoria caché en la memoria tampón de edición. Si la memoria tampón de edición llega a ser demasiado grande, el servidor de entrada de texto elimina el texto (1) más alejado de la selección actual y (2) menos recientemente modificado, y notifica al cliente de entrada de texto para actualizar el seguidor de memoria caché en la memoria tampón de edición.

En algunos ejemplos, el recurso sincroniza el contenido de la memoria tampón de edición con el contenido del control de entrada de texto de una manera que resuelva las colisiones entre los cambios simultáneos por ambas entidades. En particular, en algunos ejemplos, el recurso prioriza el contenido del control de entrada de texto al revertir los cambios en la memoria intermedia de entrada de edición que colisionan con los cambios en el control de entrada de texto. Para ello, el recurso realiza un seguimiento de los cambios que se han realizado en la memoria tampón de edición, pero que no se han confirmado por el servidor de entrada de texto como realizados en el control de entrada de texto, así como las acciones mediante teclas que dieron lugar a esos cambios. Cuando se detecta una colisión, el recurso revierte estos cambios pendientes en la memoria tampón de edición y, a continuación, reproduce las acciones mediante teclas correspondientes en el contexto de la memoria tampón de edición a medida que se actualiza con base en los cambios por parte de la aplicación.

En un ejemplo, el IME produce una secuencia de cuatro caracteres. Los tres primeros caracteres desencadenan una acción de autocorrección o autoformateo por el control de entrada de texto o la aplicación. Cuando se realiza la revisión

5 en el control de entrada de texto en respuesta a los tres primeros caracteres de la secuencia, el cliente de entrada de texto (1) omite reconocer el cuarto carácter de la secuencia del IME; (2) envía una operación de cambio de texto al servidor de entrada de texto para modificar los tres primeros caracteres de

10 la secuencia del IME en la memoria tampón de edición. Debido a que el servidor de entrada de texto recibe la operación de cambio de texto de parte del cliente de entrada de texto antes del reconocimiento del cuarto carácter de la secuencia del IME, deshace la adición del cuarto carácter en la memoria tampón de

15 edición, entonces aplica la operación de cambio de texto, a continuación, reproduce el cuarto carácter en el contexto de la operación de cambio de texto de la aplicación.

En algunos ejemplos, el proceso de aplicación y el servicio de entrada se ejecutan en el mismo sistema

20 informático. En algunos ejemplos, el proceso de aplicación y el proceso de servicio de entrada se ejecutan en diferentes sistemas informáticos. Por ejemplo, el proceso de servicio de entrada puede ejecutarse en el sistema informático del usuario, mientras el proceso de aplicación se ejecuta en un servidor en

25 la nube. Como otro ejemplo, el proceso de aplicación puede

ejecutarse en un primer sistema informático del usuario, y los procesos de servicio de entrada pueden ejecutarse tanto en el primer sistema informático del usuario como en un segundo sistema informático del usuario colaborador.

5 Al comportarse de alguna o todas las maneras descritas anteriormente, el recurso libera el proceso de solicitud de la sobrecarga de la ejecución del IME; elimina el riesgo de seguridad de ejecutar un IME del proveedor de *software* en otra aplicación del proveedor de *software*; Y permite la ejecución de
10 la aplicación y del IME en diferentes sistemas informáticos.

La figura 1 es un diagrama de bloques que muestra algunos de los componentes que pueden incorporarse en al menos algunos de los sistemas informáticos y otros dispositivos en los cuales opera el recurso. En varios ejemplos, estos
15 sistemas informáticos y otros dispositivos 100 pueden incluir sistemas informáticos servidor, sistemas informáticos de escritorio, sistemas informáticos portátiles, sistemas informáticos tipo tableta, minicomputadora portátil, teléfonos móviles, asistentes digitales personales, televisores, cámaras,
20 computadoras de automóviles, reproductores multimedia electrónicos, dispositivos de quiosco electrónico, dispositivos de mesa electrónica, dispositivos de pizarra electrónica, etcétera. En varios ejemplos, los sistemas y dispositivos informáticos pueden incluir cualquier número de los siguientes:
25 una unidad central de procesamiento (CPU, por sus siglas en

inglés) 101 para ejecutar programas de computadora; una memoria de computadora 102 para almacenar programas y datos mientras se están utilizando, que incluyen el recurso y los datos asociados, un sistema operativo que incluye un núcleo y controladores de dispositivo, y una o más aplicaciones; un dispositivo de almacenamiento permanente 103, como una unidad de disco duro o unidad *flash* para almacenar de forma permanente programas y datos; una unidad de medios legibles por computadora 104, como una unidad de disco flexible, CD-ROM o DVD, para leer programas y datos almacenados en un medio legible por computadora; y/o un subsistema de comunicaciones 105 para conectar el sistema informático a otros sistemas informáticos y/u otros dispositivos para enviar y/o recibir datos, como a través de internet u otra red alámbrica o inalámbrica y su *hardware* de red, como conmutadores, encaminadores, repetidores, cables eléctricos y fibras ópticas, emisores y receptores de luz, transmisores y receptores de radio y similares.

La figura 2 es un diagrama de flujo de datos que representa la operación del recurso en algunos ejemplos. En general, la figura 2 muestra cómo la entrada de texto generada por un dispositivo de entrada 201 se transporta a una aplicación a través de su control de edición 223.

El dispositivo de entrada recibe la entrada de usuario que especifica el texto. En varios ejemplos, un teclado físico recibe

la entrada que representa las pulsaciones de teclas; un digitalizador táctil recibe la entrada que representa los toques y/o desplazamientos en un teclado en pantalla; un micrófono u otro dispositivo de entrada de sonido recibe una entrada que representa una voz audible; una cámara u otro sensor de imagen recibe una entrada que representa la comunicación a través del lenguaje de señas americano, lectura de los labios u otro sistema de comunicación visual; etcétera.

Un IME 211 que se ejecuta en un proceso de servicio de entrada 210 recibe la entrada de especificación de texto desde el dispositivo de entrada y transforma esta entrada en un flujo de caracteres. El IME comunica la secuencia de caracteres a una memoria tampón de edición 212 que también se ejecuta en el proceso de servicio de entrada. En algunos ejemplos, la memoria tampón de edición se implementa para interactuar con el IME de la manera en la cual el control de edición de una aplicación interactúa con un IME, de manera que los IME pueden utilizarse con el recurso los cuales se diseñaron para interactuar directamente con los controles de edición.

Los caracteres que se comunican con la memoria tampón de edición por el IME se insertan en la memoria tampón de edición en un punto de inserción actual. En algunos casos, los caracteres se comunican a la memoria tampón de edición por el IME en un momento en el cual la memoria tampón de edición y el control de edición están vacíos. En tales casos, un servidor

de entrada de texto 213 se comunica con un cliente de entrada de texto 221 en el proceso de aplicación 220 para sincronizar los caracteres insertados en la memoria tampón de edición en el control de edición. Por otro lado, los caracteres también pueden comunicarse a la memoria tampón de edición por el IME en un momento en el cual la memoria tampón de edición y el control de edición contienen el texto sincronizado. Esto puede ocurrir, por ejemplo, como resultado de cualquier combinación de la memoria tampón de edición que recibió previamente el texto de parte del IME; el texto que se carga en el control de edición por la aplicación, como a partir de un documento u otro archivo; y la aplicación que altera el texto en el control de edición, como para realizar una acción de autocorrección o autoformateo, reflejar una edición por parte de un usuario colaborador, etcétera.

En algunos ejemplos, la memoria tampón de edición almacena una versión incompleta del texto contenido en el control de edición. El cliente de entrada de texto utiliza una estructura de datos de seguidor de memoria caché en memoria tampón de edición 222 para dar seguimiento a las porciones del texto en el control de edición que están presentes en la memoria tampón de edición. Cada vez que el usuario selecciona una nueva selección de texto o un punto de inserción, el recurso hace que el texto en el control de edición que rodea esa selección de texto o punto de inserción se cargue en la

memoria tampón de edición — a menos que este texto ya esté almacenado en la memoria tampón de edición — y se actualiza el seguidor de memoria caché en memoria tampón de edición. Si la memoria tampón de edición llega a ser demasiado grande, entonces el recurso elige el texto dentro de la memoria tampón de edición para descartarlo, actualizando nuevamente el seguidor de memoria caché en memoria tampón de edición. En algunos ejemplos, el proceso para elegir el texto que se descartará favorece descartar el texto que está lejos de la selección actual o punto de inserción, y que no se ha modificado recientemente.

Tal como se abordó anteriormente, es posible que el IME y la aplicación intenten simultáneamente cambiar el texto que se sincroniza entre el control de edición y la memoria tampón de edición. Cuando esto sucede, el recurso da prioridad a los cambios por la aplicación; revierte los cambios en conflicto por el IME, da efecto a los cambios realizados por la aplicación tanto en el control de edición como en la memoria tampón de edición, reproduce las acciones mediante teclas que originariamente generaron los cambios revertidos, permitiendo al IME actualizar la memoria tampón de edición usando su nuevo estado y, a continuación, propaga los nuevos cambios de memoria tampón de edición en el control de edición. Otros detalles sobre esta resolución de colisión se presentan a continuación.

Las figuras 3-6 abordadas a continuación muestran la lógica de procesamiento empleada por el recurso en varios de sus componentes en algunos ejemplos, mientras la figura 7 abordada a continuación muestra un ejemplo de las interacciones entre el servidor de entrada de texto del recurso y su cliente de entrada de texto.

La figura 3 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del IME recibidas por el servidor de entrada de texto. En 301, la instalación se divide de acuerdo con el tipo de acción del IME recibida: para las acciones de selección, el recurso continúa en 302; para las acciones de tecla abajo, el recurso continúa en 303; para las acciones de tecla arriba, el recurso continúa en 308.

En 302, donde la acción del IME es una acción de selección, el recurso envía el intervalo de selección o punto de inserción especificado por el IME al cliente de entrada de texto. Después de 302, el recurso continúa en 301 para procesar la siguiente acción del IME.

En 303, donde la acción del IME es una acción de tecla abajo, el recurso agrega el carácter de tecla especificado por el IME en la acción de tecla abajo a una lista de espera de pulsación de teclas, a partir de la cual puede reproducirse ese carácter de tecla si la inserción correspondiente del carácter en la memoria tampón de edición necesita revertirse. En 304, el

recurso envía una acción de tecla abajo al cliente de entrada de texto, cuyo procesamiento se aborda más adelante con relación a la figura 6. En 305, el recurso inserta el carácter de tecla especificado por la acción de tecla abajo en la memoria tampón de edición en el punto de inserción actual. En 306, el recurso agrega la acción de inserción de carácter realizada en 305 a una lista de espera de acciones pendientes, a partir de la cual puede revertirse la acción en caso de una colisión con un cambio de aplicación. En 307, el recurso envía una acción de inserción de carácter al cliente de entrada de texto. Después de 307, el recurso continúa en 301 para procesar la siguiente acción del IME.

En 308, donde la acción del IME es una acción de tecla arriba, el recurso envía la acción de tecla arriba al cliente de entrada de texto. Después de 308, el recurso continúa en 301 para procesar la siguiente acción del IME.

Los expertos en la técnica apreciarán que las acciones mostradas en la figura 3 y en cada uno de los diagramas de flujo abordados más adelante pueden alterarse en una variedad de maneras. Por ejemplo, el orden de las acciones puede reorganizarse; algunas acciones pueden realizarse en paralelo; las acciones mostradas pueden omitirse, u otras acciones pueden incluirse; una acción mostrada puede dividirse en subacciones o múltiples acciones mostradas pueden combinarse en una sola acción, etcétera.

La figura 4 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del cliente de entrada de texto recibidas por el servidor de entrada de texto.

5 En 401, el recurso se divide de acuerdo con el tipo de acción de cliente de entrada de texto recibido: para las acciones de reconocimiento de inserción de caracteres, el recurso continúa en 402; para las acciones de cambio de texto, el recurso continúa en 404; para las acciones de texto en memoria caché,
10 el recurso continúa en 409.

En 402, donde la acción de cliente de entrada de texto es una acción de reconocimiento de inserción de caracteres, el recurso elimina la acción de inserción de caracteres correspondiente de la lista de espera de acciones pendientes
15 del servidor de entrada de texto.

En 403, el recurso elimina el carácter correspondiente de la lista de espera de pulsaciones de teclas. Después de 403, el recurso continúa en 401 para procesar la siguiente acción de cliente de entrada de texto.

20 En 404, donde la acción de cliente de entrada de texto es una acción de cambio de texto, el recurso revierte en la memoria tampón de edición cualquier acción en la lista de espera de acciones pendientes y elimina la lista de espera de acciones pendientes. En 405, el recurso realiza la acción de
25 cambio de texto en la memoria tampón de edición. En 406, el

recurso envía un reconocimiento del cambio de texto al cliente de entrada de texto. En 407, el recurso notifica al IME que su contexto ha sido restablecido. En 408, el recurso reproduce las pulsaciones de teclas en la lista de espera de pulsaciones de teclas y vacía la lista de espera de pulsaciones de teclas. Después de 408, el recurso continúa en 401 para procesar la siguiente acción de cliente de entrada de texto.

En 409, donde la acción de cliente de entrada de texto es una acción de texto de almacenamiento en memoria caché, el recurso agrega el texto especificado por la acción de texto de almacenamiento en memoria caché en la memoria tampón de edición y ajusta el índice que el recurso mantiene en la memoria tampón de edición. En algunas modalidades, este índice está bajo el control de la memoria tampón de edición. Cuando se utiliza por el recurso, el índice de memoria tampón de edición identifica las porciones del texto en el control de edición que se almacenan en la memoria tampón de edición y las posiciones en la memoria tampón de edición en la cual se almacenan. En 410, el recurso gestiona el tamaño de la memoria tampón de edición al eliminar el texto en la memoria tampón de edición si la memoria tampón de edición ha excedido su tamaño máximo. El comportamiento del recurso en 410 se aborda con mayor detalle a continuación con relación a la figura 5. Después de 410, el recurso continúa en 401 para procesar la siguiente acción de cliente de entrada de texto.

La figura 5 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para gestionar el tamaño de la memoria tampón de edición. El recurso realiza continua a través de las acciones 501-504 mientras el tamaño de la memoria tampón de edición excede el tamaño máximo total de memoria caché. En algunos ejemplos, el recurso utiliza un tamaño máximo total de memoria caché de 200,000 caracteres. En 502, el recurso selecciona una región de memoria tampón de edición para vaciar la memoria tampón de edición. En algunos ejemplos, el recurso selecciona una región de memoria tampón de edición que está distante del punto de inserción actual y no se ha modificado recientemente. En algunos ejemplos, esto implica: dividir el texto en la memoria tampón de edición en regiones; clasificar cada región por su distancia desde el punto de inserción; clasificar cada región por el tiempo desde que se modificó por última vez; sumar la clasificación de cada región; agregar un valor, como 2, a la suma para cada región que contiene el texto introducido por parte del usuario; y seleccionar la región que tiene el resultado final más pequeño y no contiene un punto de inserción actual. En 503, el recurso elimina esta región seleccionada en 502 de la memoria tampón de edición. En 504, si el tamaño de memoria tampón de edición es aún mayor que el tamaño máximo total de memoria caché, entonces el recurso continúa en 501 para seleccionar otra

región de la memoria tampón de edición para la eliminación, de lo contrario el recurso continúa en 505. En 505, el recurso notifica al cliente de entrada de texto en cuanto a que las regiones se eliminaron para que el cliente de entrada de texto pueda actualizar el seguidor de memoria caché en memoria tampón de edición.

La figura 6 es un diagrama de flujo que muestra las acciones ejemplares que pueden realizarse por el recurso en algunos ejemplos para procesar las acciones del servidor de entrada de texto recibidas por el cliente de entrada de texto. En 601, el recurso se divide de acuerdo con el tipo de acción de servidor de entrada de texto recibida: para las acciones de intervalo de selección, el recurso continúa en 602; para las acciones eliminadas de la región de memoria tampón de edición, el recurso continúa en 604; para las acciones de tecla abajo, el recurso continúa en 605; para las acciones de tecla arriba, el recurso continúa en 607; para las acciones de inserción de caracteres, el recurso continúa en 609.

En 602, donde la acción de servidor de entrada de texto es una acción de intervalo de selección, el recurso envía el contenido del control de edición cerca del intervalo de selección especificado por la acción al servidor de entrada de texto para el almacenamiento en memoria caché en la memoria tampón de edición. En algunos ejemplos, el recurso envía 10,000 caracteres a cada lado del intervalo de selección (o punto de

inserción). En algunos ejemplos, el recurso también envía todo el intervalo de selección. En algunas modalidades, el recurso envía únicamente una porción de grandes intervalos de selección, como 50 caracteres en cada extremo del intervalo de selección. En 603, el recurso ajusta el seguidor de memoria caché en memoria tampón de edición para reflejar la adición del texto enviado en la memoria tampón de edición. Después de 603, el recurso continúa en 601 para procesar la siguiente acción de cliente de entrada de texto.

En 604, donde la acción de servidor de entrada de texto es una acción eliminada de región de memoria tampón de edición, el recurso ajusta el seguidor de memoria caché en memoria tampón de edición para reflejar la eliminación del texto especificado de la memoria tampón de edición. Después de 604, el recurso continúa en 601 para procesar la siguiente acción de servidor de entrada de texto.

En 605, donde la acción de servidor de entrada de texto es una acción de tecla abajo, el recurso transmite la acción de tecla abajo al control de edición. En 606, el recurso envía un reconocimiento de tecla abajo al servidor de entrada de texto. Después de 606, el recurso continúa en 601 para procesar la siguiente acción de servidor de entrada de texto.

En 607, donde la acción de servidor de entrada de texto es una acción de tecla arriba, el recurso transmite la acción de tecla arriba al control de edición. En 608, el recurso envía un

reconocimiento de tecla arriba al servidor de entrada de texto. Después de 608, el recurso continúa en 601 para procesar la siguiente acción de servidor de entrada de texto.

En 609, con una acción de servidor de entrada de texto está una acción de inserción de caracteres, el recurso realiza la acción de inserción de caracteres en el control de edición para insertar el carácter que especifica. En 610, si el control de edición realiza la acción de inserción de caracteres especificada, entonces el recurso continúa en 611, de lo contrario el recurso continúa en 612. En 611, el recurso envía al servidor de entrada de texto un reconocimiento de la inserción de caracteres. Después de 611, el recurso continúa en 601 para procesar la siguiente acción de servidor de entrada de texto. En 612, el recurso envía al servidor de entrada de texto una acción de cambio de texto como parte de un escenario de anulación, en el cual no se inserta nada o en el cual se insertan uno o más caracteres distintos del carácter especificado. Después de 612, el recurso continúa en 601 para procesar la siguiente acción de servidor de entrada de texto.

La figura 7 es un diagrama de comunicación que muestra un ejemplo de las interacciones entre el servidor de entrada de texto del recurso y su cliente de entrada de texto. Las comunicaciones que se mueven de izquierda a derecha se envían desde el servidor de entrada de texto 700 en el proceso de servicio de entrada al cliente de entrada de texto 750 en el

proceso de aplicación. Las comunicaciones que se mueven de derecha a izquierda se envían desde el cliente de entrada de texto al servidor de entrada de texto. En varios ejemplos, el recurso utiliza varios mecanismos de comunicación entre procesos y/o entre máquinas para intercambiar las comunicaciones mostradas en la figura 7.

Antes de enviar la primera comunicación 701, el estado del recurso es tal como se muestra a continuación en la tabla 1.

Tabla 1

10 memoria tampón de edición: im
 lista de espera de acciones pendientes:
 lista de espera de pulsaciones de teclas
 control de edición: im

En respuesta, el IME que genera un carácter de espacio,
 15 el servidor de texto aumenta la memoria tampón de edición, la
 lista de espera de acciones pendientes y la lista de espera de
 pulsaciones de teclas tal como se muestra a continuación en la
 tabla 2.

Tabla 2

20 memoria tampón de edición: im_
 lista de espera de acciones pendientes: inserción de
 caracteres ("_", 2)
 lista de espera de pulsaciones de teclas _
 control de edición: im

25 Además, en respuesta a que el IME genera un carácter de

espacio, el servidor de entrada de texto envía al cliente de entrada de texto una comunicación de tecla abajo 701 para el carácter de espacio, una comunicación de inserción de caracteres 702 para el carácter de espacio y una comunicación de tecla arriba 703 para el carácter de espacio. En respuesta a la comunicación de inserción de caracteres, el cliente de entrada de texto actualiza el control de edición para agregar el espacio tal como se muestra a continuación en la tabla 3.

Tabla 3

10 memoria tampón de edición: im_

 lista de espera de acciones pendientes: inserción de

 caracteres (" ", 2)

 lista de espera de pulsaciones de teclas _

 control de edición: im_

15 El cliente de entrada de texto también reconoce la

 comunicación de tecla abajo 701 con un reconocimiento de tecla

 abajo 751, la comunicación de inserción de caracteres 702 con

 un reconocimiento de inserción de caracteres 752 y la

 comunicación de tecla arriba 703 con un reconocimiento de

20 tecla arriba 753. En respuesta al reconocimiento de inserción

 de caracteres, el servidor de entrada de texto elimina la

 inserción de espacio de la lista de espera de acciones

 pendientes y el carácter de espacio de la lista de espera de

 pulsaciones de teclas tal como se muestra a continuación en la

25 tabla 4.

Tabla 4

memoria tampón de edición: im_

lista de espera de acciones pendientes:

lista de espera de pulsaciones de teclas

5 control de edición: im_

En el procesamiento de la acción de inserción de caracteres, la aplicación realiza una operación de autocorrección al texto en el control de edición tal como refleja la siguiente Tabla 5, lo cual transforma "im" en "l'm".

10 **Tabla 5**

memoria tampón de edición: im_

lista de espera de acciones pendientes:

lista de espera de pulsaciones de teclas

control de edición: l'm_

15 Mientras la aplicación estaba realizando la operación de autocorrección, el servidor de entrada de texto recibió el siguiente carácter, "h", emitido por el IME. En respuesta a la recepción del carácter "h", el servidor de entrada de texto actualizó la memoria tampón de edición, la lista de espera de acciones pendientes y la lista de espera de pulsaciones de

20 teclas tal como se muestra a continuación en la tabla 6.

Tabla 6

memoria tampón de edición: im_h

lista de espera de acciones pendientes: inserción de

25 caracteres ("h", 3)

lista de espera de pulsaciones de teclas h

control de edición: l'm_

Además, en respuesta a que el IME genera el carácter "h", el servidor de entrada de texto envió las comunicaciones de tecla abajo, inserción de caracteres y tecla arriba 704-706 para el carácter "h".

Debido a la operación de autocorrección por la aplicación en respuesta a la comunicación de inserción de caracteres para el carácter de espacio, en lugar de responder a las comunicaciones 704-706 con respecto al carácter "h", el cliente de entrada de texto en su lugar envía una comunicación de cambio de texto 754 que especifica la sustitución de los caracteres desde la posición 0 hasta la posición 2 en la memoria tampón de edición ("im") por "l'm".

En respuesta, el servidor de entrada de texto procede de la siguiente manera: en primer lugar, el servidor de entrada de texto revierte las acciones mostradas en la lista de espera de acciones pendientes en la memoria tampón de edición tal como se muestra a continuación en la tabla 7.

Tabla 7

memoria tampón de edición: im_

lista de espera de acciones pendientes:

lista de espera de pulsaciones de teclas h

control de edición: l'm_

A continuación, el servidor de entrada de texto aplica la

acción de cambio de texto en la comunicación 752, tal como se muestra a continuación en la tabla 8.

Tabla 8

memoria tampón de edición: l'm_

5 lista de espera de acciones pendientes:

lista de espera de pulsaciones de teclas h

control de edición: l'm_

El servidor de entrada de texto envía una comunicación de reconocimiento de cambio de texto 707 y reproduce los
 10 caracteres de la lista de espera de pulsación de teclas en el estado actual de la memoria tampón de edición tal como se muestra a continuación en la tabla 9. La reproducción de la lista de espera de pulsaciones de teclas agrega una acción de inserción de caracteres a la lista de espera de acciones
 15 pendientes. La acción de inserción de caracteres permanece en la lista de espera de acciones pendientes y el carácter "h" permanece en la lista de espera de pulsaciones de teclas hasta que el cliente de entrada de texto reconoce una comunicación de inserción de caracteres de parte del servidor de entrada de
 20 texto para el carácter "h".

Tabla 9

memoria tampón de edición: l'm_h

lista de espera de acciones pendientes: inserción de
 caracteres ("h", 4)

25 lista de espera de pulsaciones de teclas h

control de edición: l'm_

Como parte de la reproducción, el servidor de entrada de texto envía las comunicaciones de tecla abajo, inserción de caracteres y tecla arriba de cliente de entrada de texto 708-710 para el carácter "h". En respuesta a la comunicación de inserción de caracteres, el cliente de entrada de texto actualiza el control de edición para agregar el carácter "h" tal como se muestra a continuación en la tabla 10. La acción de inserción para el carácter "h" permanece en la lista de espera de acción pendiente y el carácter "h" permanece en la lista de espera de pulsación de teclas hasta que la acción de inserción de caracteres para el carácter "h" se reconozca por el cliente de entrada de texto.

Tabla 10

memoria tampón de edición: l'm_h

lista de espera de acciones pendientes ("h", 4)

lista de espera de pulsaciones de teclas h

control de edición: l'm_h

Finalmente, el cliente de entrada de texto envía los reconocimientos 755-757 para las acciones de tecla abajo, inserción de caracteres y tecla arriba para el carácter "h". En respuesta al reconocimiento de inserción de caracteres, el servidor de entrada de texto vacía la lista de espera de acciones pendientes y la lista de espera de pulsaciones de teclas tal como se muestra a continuación en la tabla 11.

Tabla 11

memoria tampón de edición: l'm_h

lista de espera de acciones pendientes:

lista de espera de pulsaciones de teclas

control de edición: l'm_h

5 Los expertos en la técnica apreciarán que las
comunicaciones mostradas en la figura 7 pueden alterarse de
una variedad de maneras. Por ejemplo, las comunicaciones
pueden realizarse en órdenes diferentes; diferentes tipos de
comunicación pueden sustituirse por los tipos de comunicación
10 mostrados; pueden utilizarse más o menos tipos de
comunicación; una comunicación mostrada puede dividirse en
subcomunicaciones, o múltiples comunicaciones mostradas
pueden combinarse en una única comunicación; etcétera.

En algunos ejemplos, el recurso proporciona un sistema
15 informático para proporcionar la salida textual de un editor de
método de entrada a una aplicación. El sistema informático
comprende: una primera región de memoria configurada para
almacenar una versión de un contenido de texto al cual se
aplica la salida textual del editor de método de entrada; una
20 segunda región de memoria configurada para almacenar una
versión de un contenido de texto al cual se aplican las
modificaciones de texto realizadas por la aplicación; y un
subsistema de unificación configurado para unificar los
contenidos de la primera y segunda región de memoria de una
25 manera que favorezca el contenido de la segunda región de

memoria.

En algunos ejemplos, el recurso proporciona un medio legible por computadora que tiene el contenido configurado para provocar que un sistema informático procese la entrada textual generada con un dispositivo de entrada de usuario: almacene la
5 entrada textual en una primera estructura de datos contenida en el espacio de dirección de un primer proceso; y sincronice el contenido de la primera estructura de datos a una segunda estructura de datos contenida en el espacio de dirección de un
10 segundo proceso distinto del primer proceso, una aplicación que se ejecuta en el segundo proceso se configura para consumir una entrada textual sincronizada en la segunda estructura de datos.

En algunos ejemplos, la instalación proporciona un medio legible por computadora que almacena una estructura de datos
15 de sincronización técnica con relación a una primera memoria tampón a la cual se han aplicado tanto (1) la entrada de usuario textual como (2) los cambios de texto de aplicación sincronizados desde una segunda memoria tampón. La estructura de datos comprende: una o más primeras entradas
20 que contienen cada una la información que identifica una acción de edición que (a) se ha aplicado a la primera memoria tampón en respuesta a la entrada textual de usuario, (b) se ha comunicado a la segunda memoria tampón y (c) no se ha reconocido por la segunda memoria tampón, de tal manera que
25 el contenido de las primeras entradas sea utilizable para

revertir las acciones de edición que se han aplicado a la primera memoria tampón y sean incompatibles con los cambios de texto de aplicación en la segunda memoria tampón.

Los expertos en la técnica apreciarán que el recurso
5 anteriormente descrito puede adaptarse o extenderse directamente de varias maneras. Aunque la descripción anterior hace referencia a las modalidades particulares, el alcance de la invención se define únicamente por las siguientes reivindicaciones y los elementos citados en las mismas.

10

15

20

25

