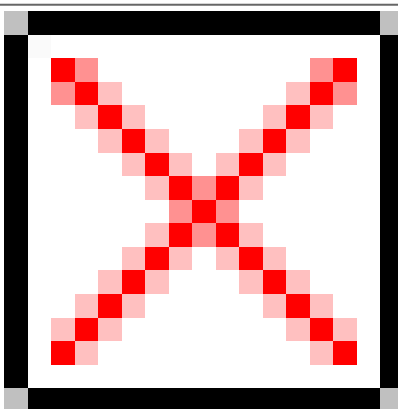


DIRECCIÓN DE NUEVAS CREACIONES

SOLICITUD FASE NACIONAL - PCT



1	Título de la Invención (200 caracteres o espacios máximos)		
COMPILACION DE TRANSFORMACION EN UNA INTERFASE DE USUARIO DE RECALCULO			
2	Datos del Solicitante / Titular		
Nombre:	MICROSOFT TECHNOLOGY LICENSING, LLC	Dirección Electrónica:	infopia@cardenasycardenas.com
Dirección:	One Microsoft Way 98052	Domicilio/País de constitución:	ESTADOS UNIDOS DE AMÉRICA - WASHINGTON - REDMOND
Identificación:			
☐ CEDULA DE CIUDADANIA ☒ EMPRESA EXTRANJERA ☐ PASAPORTE		☐ CEDULA DE EXTRANJERIA ☐ NIT	

Número:		546817687188-	
3	Solicitantes		
Apellidos - Nombres o Razón Social		Tipo	Identificación
1.	MICROSOFT TECHNOLOGY LICENSING, LLC	NN	546817687188
4	Datos del Inventor		
Nombre:	Andrew Douglas REDDISH	Dirección Electrónica:	infopi@cardenasycardenas.com
Dirección:	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399	Domicilio/País de constitución:	ESTADOS UNIDOS DE AMÉRICA - WASHINGTON - REDMOND
Identificación:			
☐ CEDULA DE CIUDADANIA ☐ EMPRESA EXTRANJERA ☐ PASAPORTE ☐ CEDULA DE EXTRANJERIA ☐ NIT ☒ No Aplica			
Número:	-		
5	Inventor(es)		
Apellidos - Nombres		Domicilio	
1.	REDDISH Andrew Douglas	ESTADOS UNIDOS DE AMÉRICA	
2.	COLLE Olivier	ESTADOS UNIDOS DE AMÉRICA	
3.	GRUIAN Radu B	ESTADOS UNIDOS DE AMÉRICA	
4.	ANUAR Nizam	ESTADOS UNIDOS DE AMÉRICA	
5.	SARKAR Jaideep	ESTADOS UNIDOS DE AMÉRICA	
6.	MITAL Vijay	ESTADOS UNIDOS DE AMÉRICA	

6	Datos Inventor(es)			
	País de Residencia	Departamento/Estado	Ciudad	Dirección
1.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
2.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
3.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
4.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
5.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
6.	ESTADOS UNIDOS DE AMÉRICA	WASHINGTON	REDMOND	c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, 98052-6399
7	Datos del Representante Legal / Apoderado			

Nombre:	LUZ HELENA ADARVE GOMEZ	Dirección Electrónica:	infopi@cardenasycardenas.com
Dirección:	Cra 7 No 71 - 52 Torre B Piso 9	Domicilio/País de constitución:	COLOMBIA - BOGOTA D.C. - BOGOTA D.C.
Identificación: ☒ CEDULA DE CIUDADANIA ☐ CEDULA DE EXTRANJERIA ☐ EMPRESA EXTRANJERA ☐ NIT ☐ PASAPORTE Número: 41575435-			
Presentación de Poder			
Año de Radicación			
Número de Radicación			
8	Datos Solicitud: PCT / WO		
Número Solicitud:	pct/us2014/033708	Fecha Solicitud:	11/04/2014
Número Publicacion:	WO 2014/169160	Fecha Publicacion:	16/10/2014
9	Declaraciones de prioridad	☒ SI ☐ NO	
1.	(33) País de origen ESTADOS UNIDOS DE AMÉRICA	Codigo del país US	(31) No. Solicitud 13/862,277 (32) Fecha 12/04/2013
10	Reivindicaciones		
Número reivindicaciones:		10	Pago Reivindicaciones: No
11	Reducción de tasas.		
<i>Declaro que carezco de medios económicos para presentar la solicitud de patente.</i> ☐ SI ☒ NO Nota: En caso de ser persona natural y carecer de medios económicos, y por lo tanto, aplique la reducción de tasas a que se refiere la resolución vigente en tarifas, debe firmar la presente solicitud bajo la gravedad de juramento.			
☐ Micro, pequeñas y medianas empresas ☐ Universidades públicas o privadas			

☐ Entidades sin ánimo de lucro

Debe aportar los documentos que se indican en el numeral 17 de anexos

12	Documentos Anexos
-----------	--------------------------

- | |
|--|
| <ul style="list-style-type: none">☒ Reivindicaciones☒ Descripción☒ Dibujos y/o Figuras☒ Resumen☐ Certificado Depósito Material Biológico☐ Uso de Conocimiento tradicional☐ Listado de secuencias☒ Artes finales 12 x 12 cm☒ Poderes, si fuere el caso☐ Copia de la primera solicitud si se reivindica prioridad☐ Traducción simple de la primera solicitud, si se reivindica prioridad☒ Otros Anexos |
|--|

P-018

From the INTERNATIONAL BUREAU

PCTNOTIFICATION OF THE RECORDING
OF A CHANGE(PCT Rule 92bis.1 and
Administrative Instructions, Section 422)

To:

MICROSOFT TECHNOLOGY LICENSING, LLC
Attn - Linda Y. Johnson (ljohnson)
LCA - International Patents
One Microsoft Way - 8/1174
Redmond, Washington 98052-6399
ÉTATS-UNIS D'AMÉRIQUE

Date of mailing (day/month/year) 16 July 2015 (16.07.2015)	
Applicant's or agent's file reference 339064-02	IMPORTANT NOTIFICATION
International application No. PCT/US2014/033708	International filing date (day/month/year) 11 April 2014 (11.04.2014)

1. The following indications appeared on record concerning:			
☒ the applicant ☐ the inventor ☐ the agent ☐ the common representative			
Name and Address MICROSOFT CORPORATION One Microsoft Way Redmond, Washington 98052-6399 United States of America		State of Nationality US	State of Residence US
		Telephone No.	
		Facsimile No.	
		E-mail address	
2. The International Bureau hereby notifies the applicant that the following change has been recorded concerning:			
☒ the person ☐ the name ☐ the address ☐ the nationality ☐ the residence			
Name and Address MICROSOFT TECHNOLOGY LICENSING, LLC One Microsoft Way Redmond, WA 98052 United States of America		State of Nationality US	State of Residence US
		Telephone No.	
		Facsimile No.	
		E-mail address ☐ Notifications by e-mail authorized	
3. Further observations, if necessary: The address for correspondence has been changed accordingly, as indicated in the addressee box at the top of this form.			
4. A copy of this notification has been sent to:			
☒ the receiving Office ☐ the International Searching Authority ☐ the Authority(ies) specified for supplementary search		☒ the International Preliminary Examining Authority ☐ the designated Offices concerned ☒ the elected Offices concerned ☐ other:	

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland	Authorized officer Blanc Veronique e-mail pt02.pct@wipo.int Telephone No. +41 22 338 74 02
---	--

Facsimile No. +41 22 338 82 70

Form PCT/IB/306 (January 2009)

1/Z6L6PDWCPXU8

UNITED STATES OF AMERICA

001226

The State of Washington



Secretary of State

APOSTILLE

(Convention de la Haye du 5 Octobre 1961)

- 1. Country: **United States of America**
- 2. This public document has been signed by: ANGELA L. WALKER
- 3. acting in the capacity of: Notary Public, state of Washington
- 4. bears the seal/stamp of: ANGELA L. WALKER

CERTIFIED

- 5. at: Olympia, Washington
- 6. the: 29 day of October, 2014
- 7. by: Kim Wyman, Secretary of State
- 8. No: 201417782
- 9. Seal/Stamp:
- 10. Signature:



Given under my hand and the Seal of the State of Washington at Olympia, the State Capital

Kim Wyman, Secretary of State

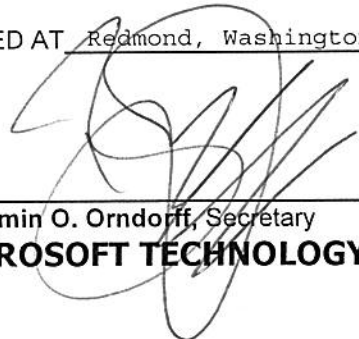
- Colombia -

POWER OF ATTORNEY

In the city of Redmond, Washington (USA) on the day of the month of October, 2014 **Benjamin O. Orndorff**, of age, acting in his capacity of Secretary and legal representative officer of **MICROSOFT TECHNOLOGY LICENSING, LLC**, legally constituted and in existence in accordance with the laws of the State of Washington, United States of America, and domiciled at One Microsoft Way, Redmond, Washington 98052, United States of America, grants hereby special, ample and sufficient Power of Attorney to DARIO CARDENAS-NAVAS and/or LUZ HELENA ADARVE-GOMEZ and/or EDUARDO CARDENAS-CABALLERO and/or BERNARDO P. CARDENAS-MARTINEZ and/or JUANITA ACOSTA GOMEZ and/or ANA MARIA RUIZ RUEDA, registered lawyers of the firm CARDENAS & CARDENAS-LAWYERS, domiciled at Carrera 7 No. 71-52, Torre B, Piso 9 Bogotá, D.C., Colombia, to carry before the corresponding Colombian National Offices and authorities the proceedings to obtain trademarks, patents, utility models, industrial drawings and designs, deposits of commercial names and ensigns, as well as processing of renewals, assignments, change of name or domicile of the owner, waivers and licenses of the same for which purpose they are empowered to take before said authorities all necessary actions for the indicated purposes; to file applications, statements, claims, formulate descriptions, amendments, oppositions and appeals; to file nullity actions or act as third interested party before the Council of State, or act before any other competent court; pay all taxes and installments and make any other payments determined by law; to receive every document and values issuing the respective discharge; to fulfill any other requirements and take, in general, every measure that they shall deem opportune for the defense and protection of the Industrial Property of the company in the event of infringements, oppositions, proceedings for cancellation or annulment, imposition of fines, passing the antecedents to the courts; they are empowered to appear as plaintiff or defendant before the competent judges, being entitled to transact, submit to arbitration, waive, receive, resign, withdraw, appeal and all other legal powers which may be necessary, and it is hereby declared ever since now valid and good whatever such attorneys do for its benefit, granting them also power to substitute this proxy and if necessary to revoke such substitutions.

That for purposes of articles 597 and 543 of the Colombian code of Commerce the attorneys are empowered to receive service of notice and designate judicial and extrajudicial attorneys.

SIGNED AT Redmond, Washington (USA) ON October 14, 2014

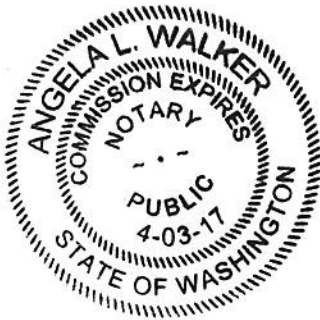
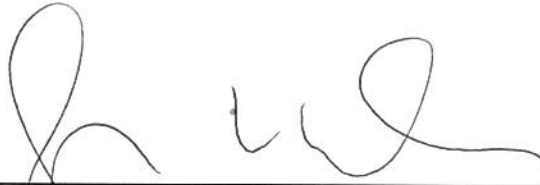
BY 
Benjamin O. Orndorff, Secretary

MICROSOFT TECHNOLOGY LICENSING, LLC.

STATE OF WASHINGTON)
) ss:
COUNTY OF KING)

I certify that I know or have satisfactory evidence that Benjamin O. Orndorff is the person who appeared before me, and said person acknowledged that he signed this instrument, on oath stated that he was authorized to execute the instrument and acknowledged it as the Secretary of Microsoft Technology Licensing, LLC to be the free and voluntary act of such party for the uses and purposes mentioned in the instrument.

Dated this 14th day of October 2014



ANGELA L. WALKER - NOTARY PUBLIC
In and for the State of Washington USA.
Residing at Seattle, Washington USA.
My Appointment expires: April 3, 2017


Washington
 Oficina de la Secretaría de Estado
 Inválido si se Retira

ESTADOS UNIDOS DE AMÉRICA

El Estado de Washington



Secretaria de Estado

Apostilla

(Convención de la Haya de 5 octubre 1961)

1. País: **Estados Unidos de América**
2. Este documento público
ha sido firmado por **ANGELA L. WALKER**
3. que actúa en la calidad de: **Notario Público, Estado de Washington**
4. lleva el sello/timbre de **ANGELA L. WALKER**

CERTIFICA

5. en Olympia, Washington
6. el 29 de octubre de 2014
7. por Kim Wyman, Secretaria de Estado
8. No. 201417782
9. Sello/timbre:



10. Firma:

Expedido bajo mi firma y Sello del Estado
de Washington en Olympia, la Capital de
Estado

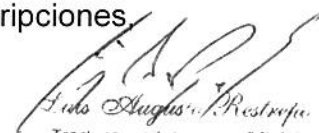


Kim Wyman, Secretaria de Estado

-Colombia-

PODER DE ABOGADO

En la ciudad de Redmond, Washington, Estados Unidos el día del mes de octubre de dos mil catorce (2014), **Benjamin O. Orndorff**, mayor de edad, actuando en calidad de Secretario y representante legal de **MICROSOFT TECHNOLOGY LICENSING, LLC**, una sociedad legalmente constituida y en vigencia de conformidad con las leyes del Estado de Washington, Estados Unidos de América y con domicilio en One Microsoft Way, Redmond, Washington, 98052, Estados Unidos de América, otorga por el presente Poder especial, amplio y suficiente a DARIO CARDENAS-NAVAS y/o LUZ HELENA ADARVE GOMEZ y/o EDUARDO CARDENAS-CABALLERO y/o BERNARDO P. CARDENAS-MARTINEZ y/o JUANITA ACOSTA GOMEZ y/o ANA MARÍA RUIZ RUEDA, abogados registrados de la firma CARDENAS & CARDENAS-LAWYERS, con domicilio en la carrera 7 No. 71-52, torre B, piso 9, Bogotá D.C., Colombia, para realizar ante las autoridades y las oficinas nacionales Colombianas correspondientes los procesos para obtener marcas registradas, patentes, modelos de utilidad, diseños y dibujos industriales, depósitos de insignias y nombres comerciales, así como también procesar renovaciones, cesiones, cambios de nombre o domicilio del propietario, renunciaciones y licencias de las mismas para cuyo propósito ellos están facultados para tomar ante dichas autoridades todas las acciones necesarias para los propósitos indicados; para presentar solicitudes, declaraciones, reclamaciones, formular descripciones,


Luis Augusto Restrepo
Traductor e Interprete Oficial
Licencia No. 1725 Ministerio de
Relaciones Exteriores

correcciones, oposiciones y apelaciones; pagar todos los impuestos y cuotas y hacer cualesquier otros pagos determinados por ley; para recibir todos los documentos y valores que emiten la respectiva cancelación; para cumplir cualesquier otros requerimientos y tomar, en general, todas las medidas que lleguen a ser oportunas para la defensa y protección de la Propiedad Industrial de la Compañía en el evento de infracciones, oposiciones, procedimientos para cancelación o anulación, imposición de multas, paso de antecedentes a los tribunales; ellos están facultados para comparecer como demandante o defensor ante los jueces competentes, tienen derecho a transar, presentar arbitraje, renunciar, recibir, renunciar, retirar, apelar y todas las otras facultades legales que puedan ser necesarias, y por la presente declaro desde ahora válido y bueno todo lo que dichos abogados hagan para su beneficio, otorgándoles también facultad para sustituir este poder y si es necesario revocar dichas sustituciones.

Que para propósitos de los artículos 597 y 543 del Código de Comercio Colombiano los abogados tienen facultad para recibir notificaciones judiciales y designar abogados judiciales y extrajudiciales.

FIRMADO EN Redmond, Washington, Estados Unidos el 14 de octubre de 2014.

POR: (aparece firma ilegible)

Benjamin O. Orndorff, Secretario

MICROSOFT TECHNOLOGY LICENSING, LLC.


Lyla Auguste Restrepo
Traductor e Interprete Oficial
Licencia No. 1725 Ministerio de
Relaciones Exteriores

Lyra Augusta Restrepo
Traductor e intérprete Oficial
Licencia No.1725 Ministerio de
Relaciones Exteriores

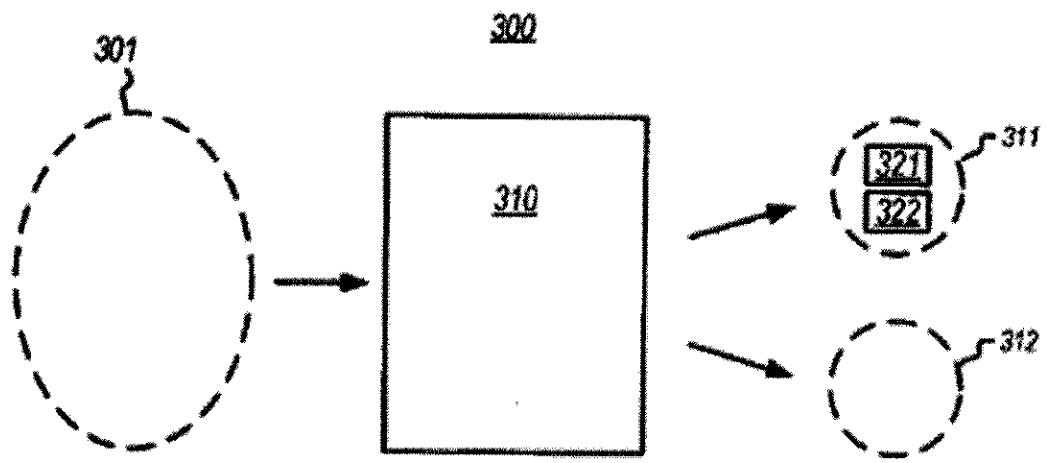
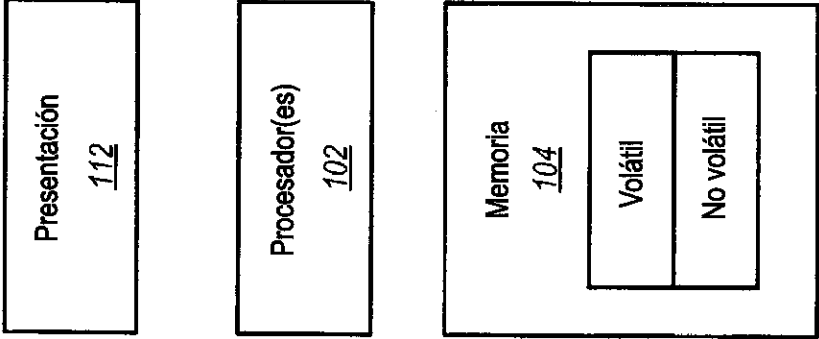


FIG. 3

RESUMEN

La complicación de una cadena de transformación de una interfase de usuario de recálculo que presenta una lona electrónica
5 que contiene uno o más resultados presentados de una cadena de transformación. La cadena de transformación incluye transformaciones entre una fuente de datos respectiva y un colector de datos. La edición de usuario de la interfase de usuario de recálculo puede hacer que una o más de las transformaciones se
10 vuelvan a ejecutar, ocasionando así el recálculo. La compilación involucra analizar la cadena de transformación de la interfase de usuario de recálculo para dependencias para crear una gráfica de dependencia de dependencias entre entidades. Por ejemplo, algunas dependencias pueden ser entre entidades con el fin de indicar que si
15 una entidad es evaluada, entonces la otra también debe ser evaluada. La gráfica de dependencia entonces se utiliza para crear un nivel inferior de pasos de ejecución. La gráfica de dependencia además se proporciona a un tiempo de ejecución para el programa, de manera que la gráfica de dependencia puede estar disponible
20 durante la operación de la interfase de usuario de recálculo.



Sistema de Cómputo
100

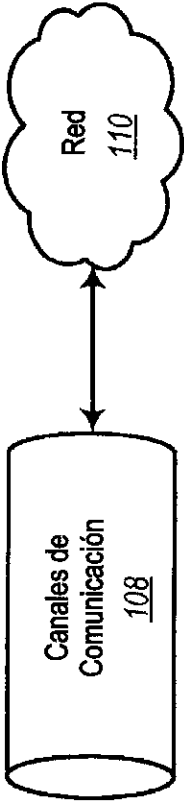


FIG. 1

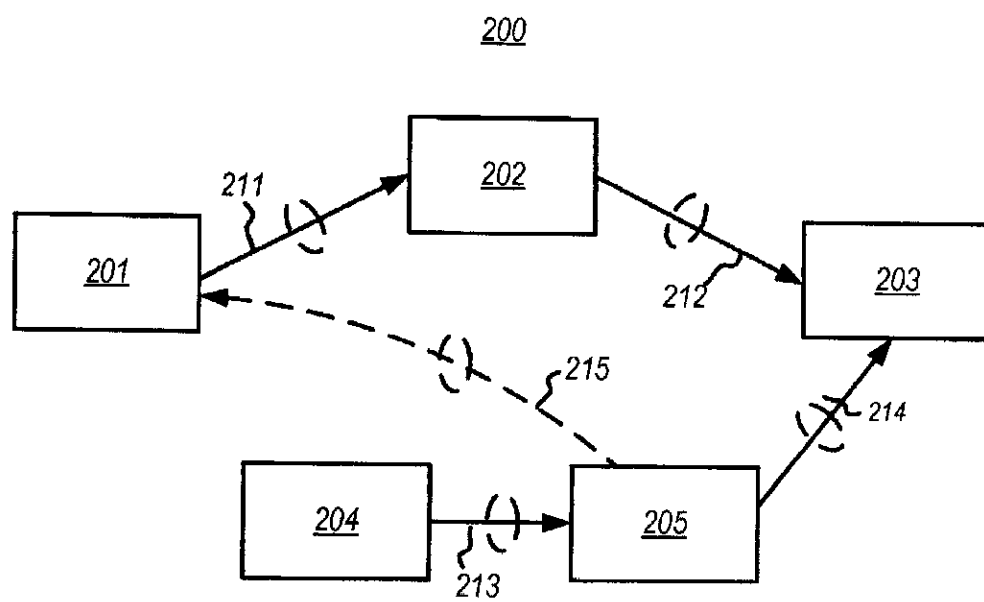


FIG. 2

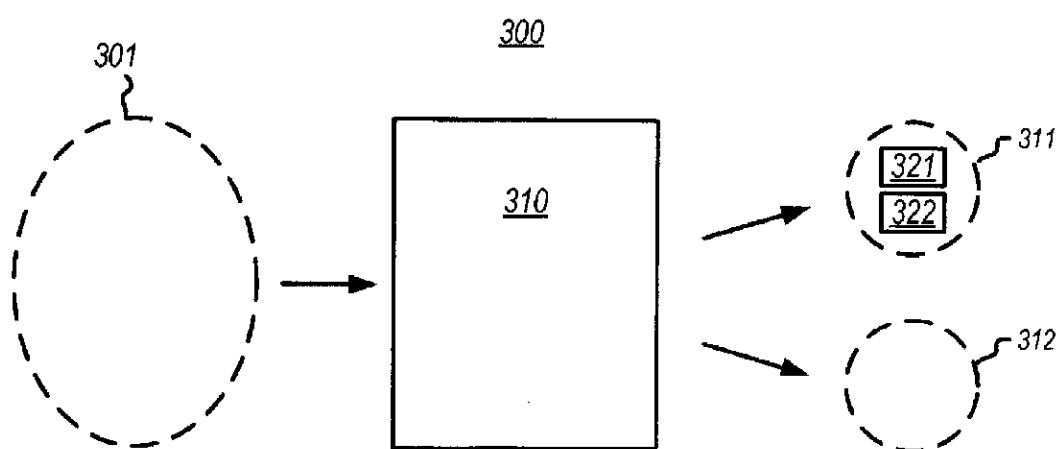


FIG. 3

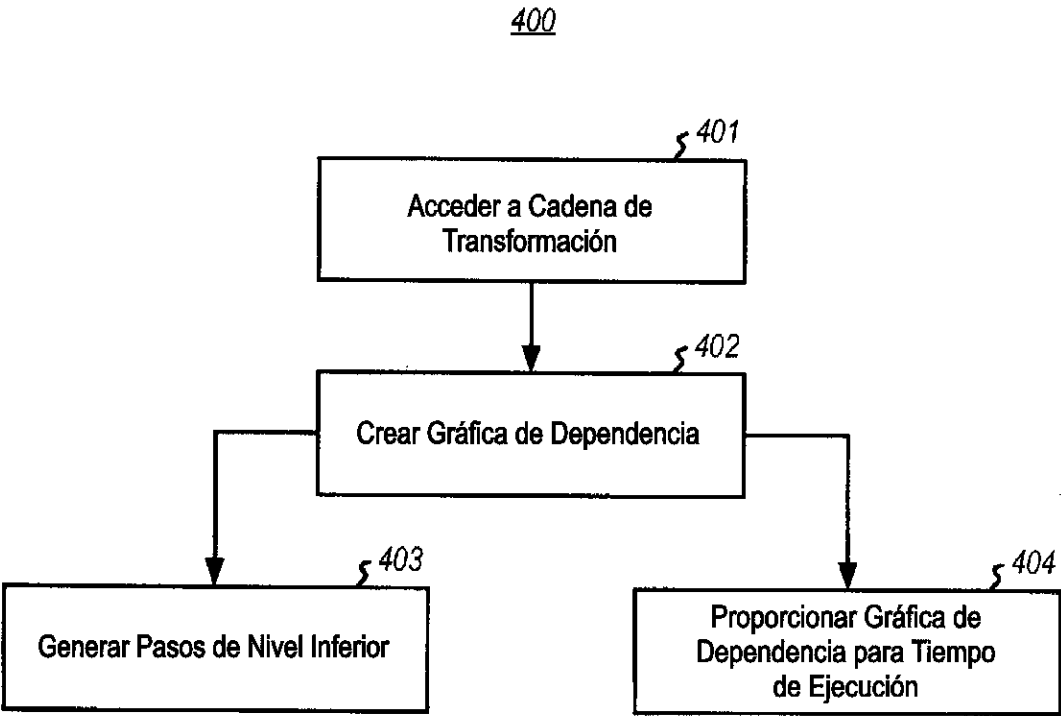


FIG. 4

500

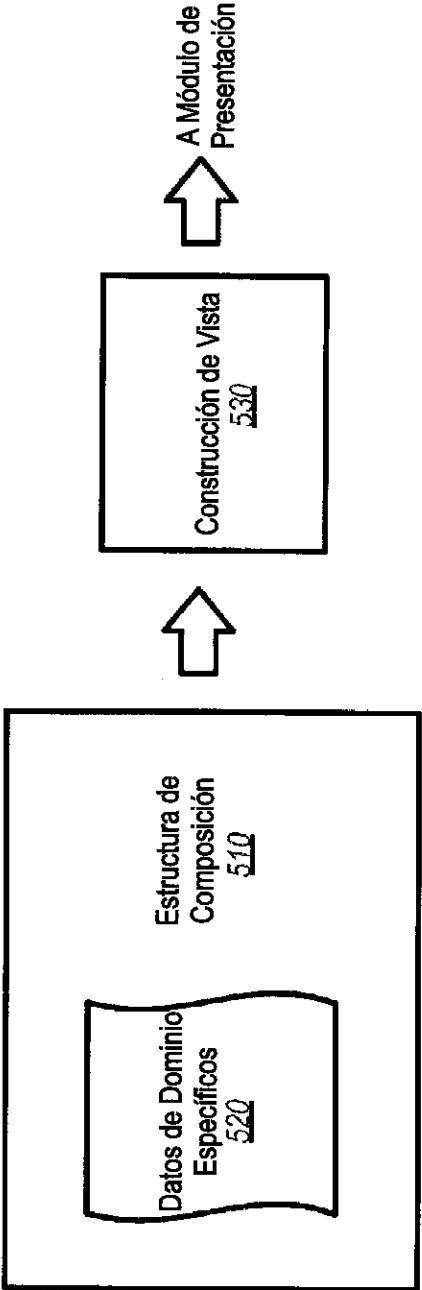


FIG. 5

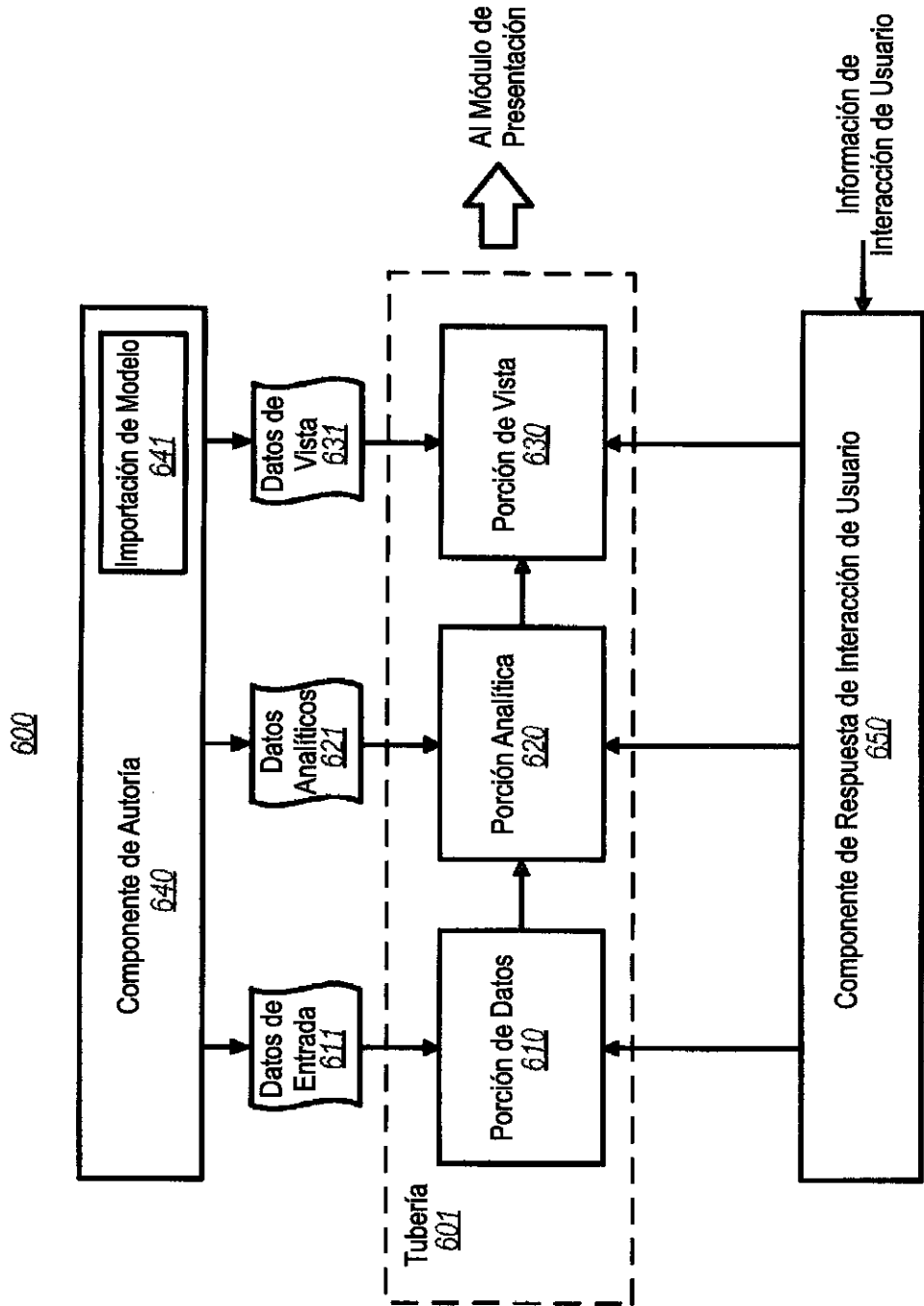


FIG. 6

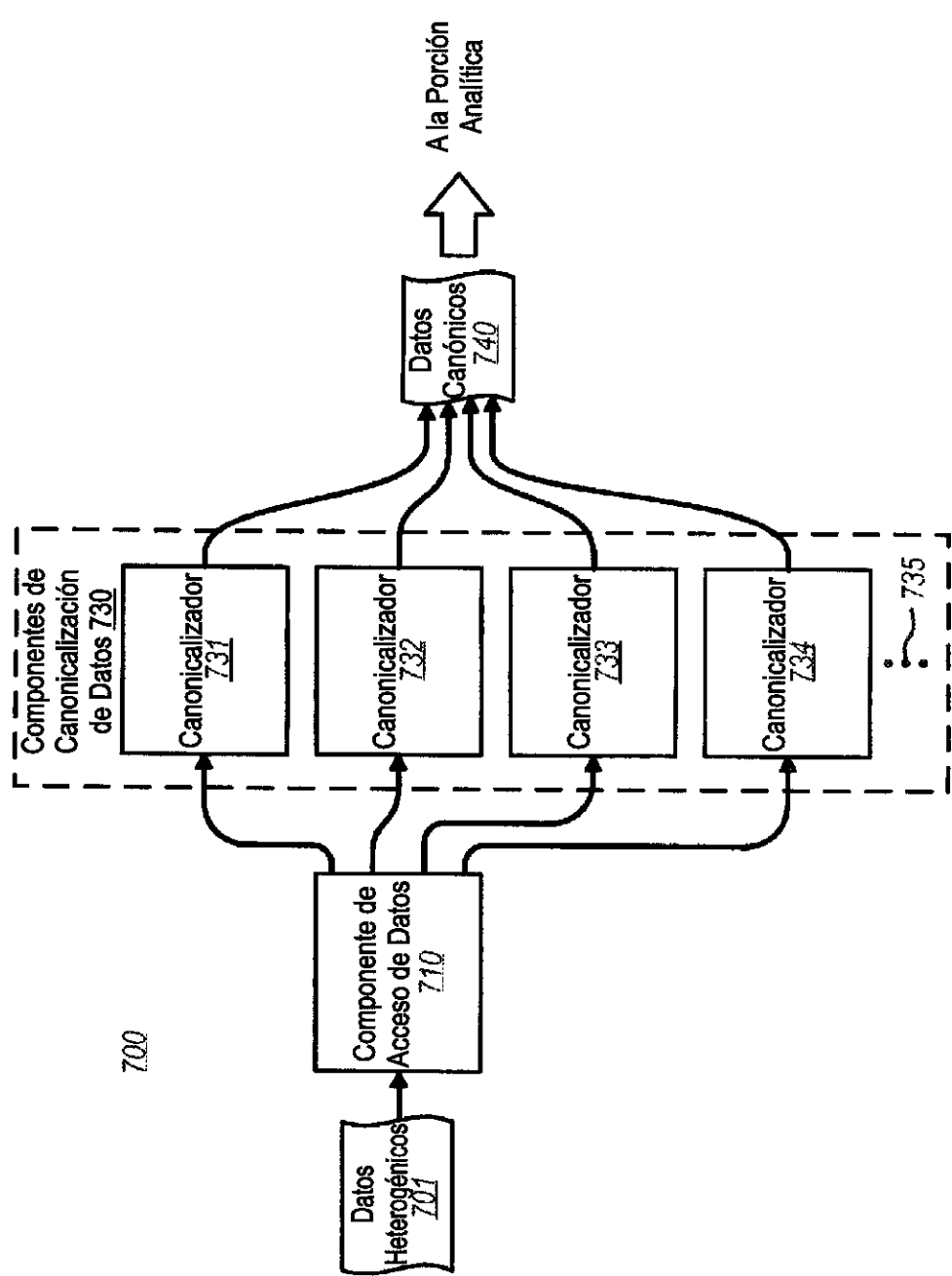


FIG. 7

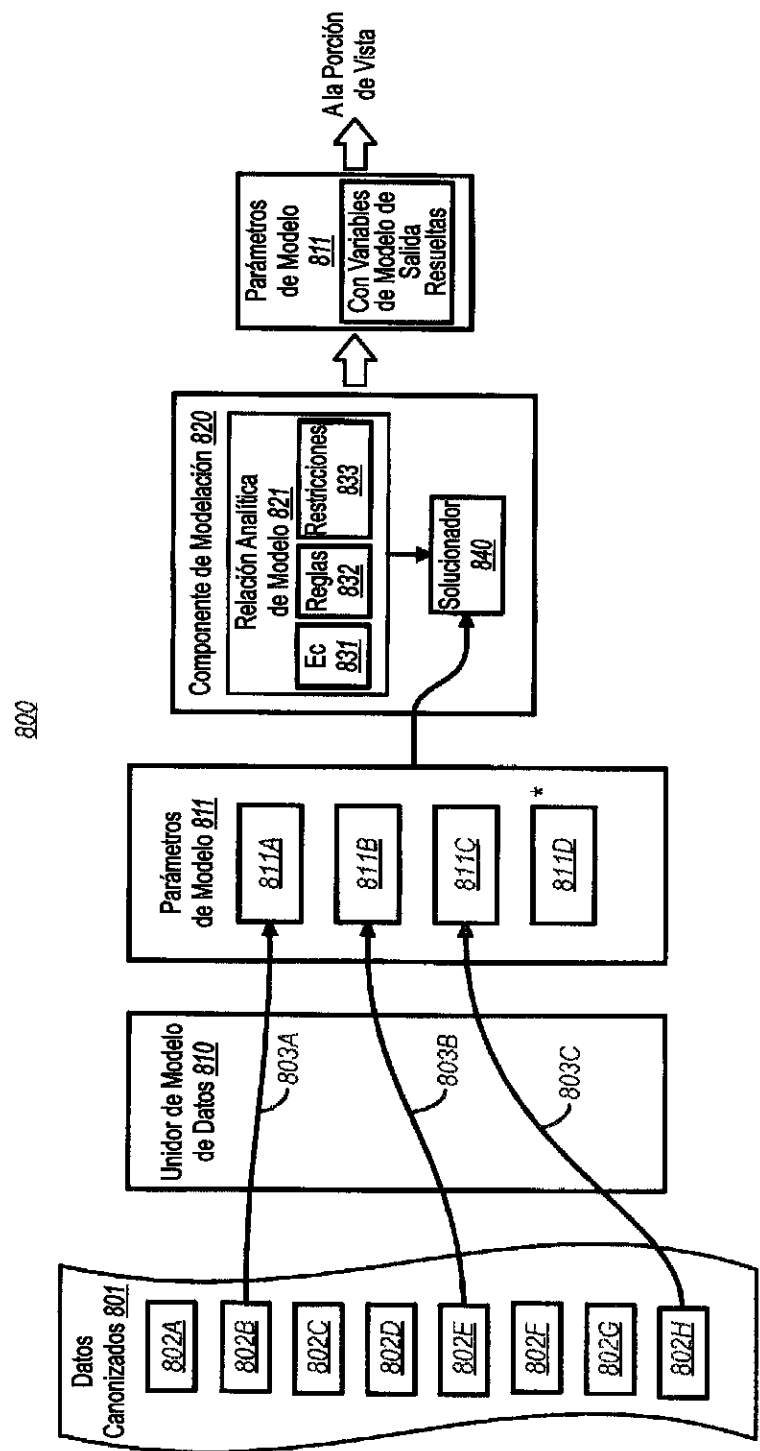


FIG. 8

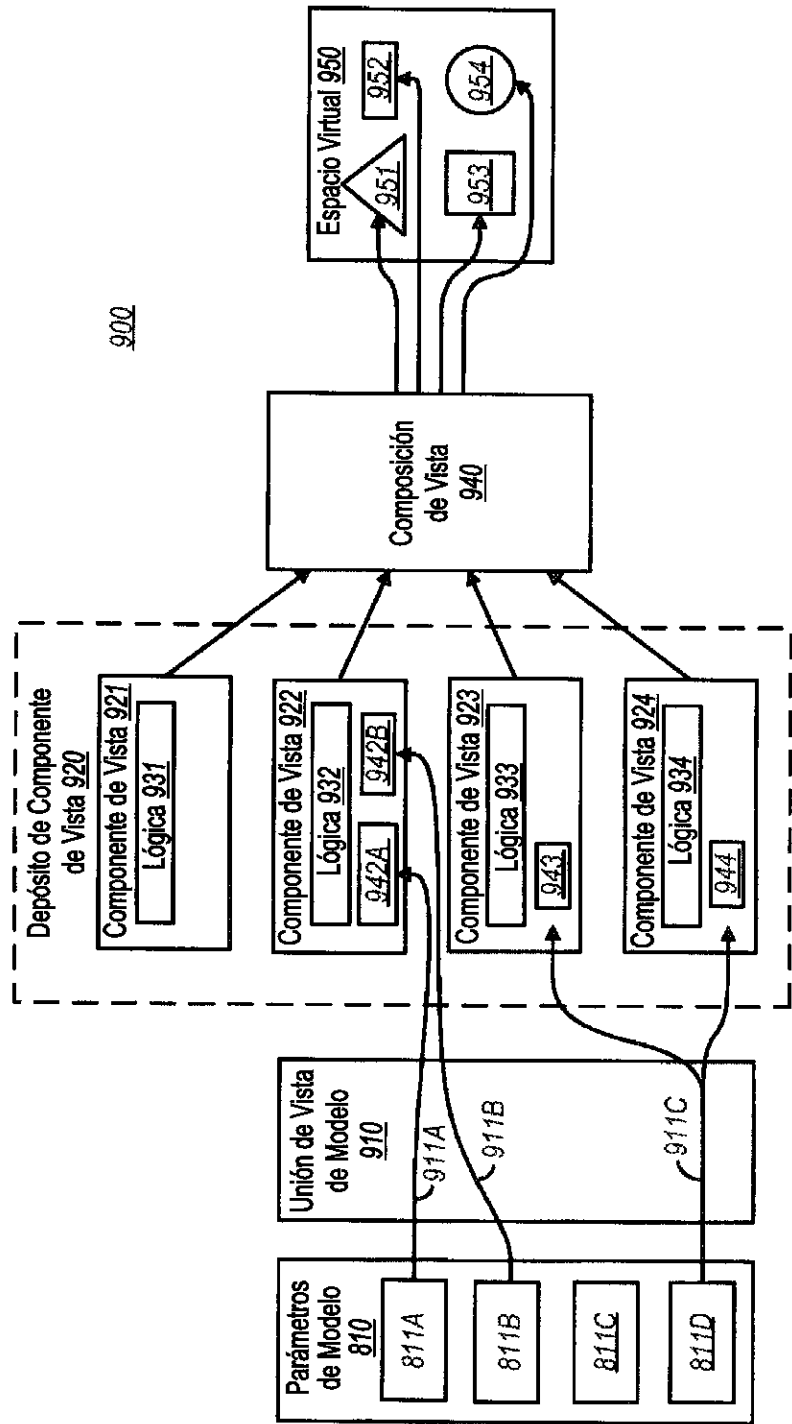


FIG. 9

COMPILACION DE TRANSFORMACION EN UNA INTERFASE DE USUARIO DE RECALCULO

ANTECEDENTES

5

Un documento de recálculo es un documento que muestra varias fuentes de datos y colectores de datos, y permite una transformación declarativa entre una fuente de datos y un colector de datos. Para un grupo dado de varias fuentes de datos y colectores de datos de interconexión de transformaciones, la salida de la fuente de datos puede ser consumida por el colector de datos, o la salida de la fuente de datos puede ser sometida a transformaciones antes de ser consumida por el colector de datos. Estas varias transformaciones son evaluadas dando como resultado una o más salidas representadas a través del documento de recálculo.

15

El usuario puede agregar y editar las transformaciones declarativas sin tener conocimiento de codificación en profundo. Dicha edición automáticamente hace que las transformaciones sean recalculadas, ocasionando un cambio en una de más salidas.

20

Un ejemplo específico de un documento de recálculo es un documento de hoja de cálculo, el cual incluye una rejilla de celdas. Cualquier celda dada puede incluir una expresión que es evaluada para producir un valor particular que es presentado en la celda. La expresión puede referirse a una fuente de datos, tal como una o más de otras celdas o valores.

25

BREVE DESCRIPCION DE LA INVENCION

Al menos algunas modalidades aquí descritas se refieren a la compilación de una cadena de transformación de una interfase de usuario de recálculo. La cadena de transformación incluye una transformación declarativa entre fuentes de datos respectivas y colectores de datos. Por ejemplo, en el contexto de una hoja de cálculo, el colector de datos puede ser una celda de hoja de cálculo particular, la transformación puede ser la expresión asociada con la celda particular, y la fuente de datos puede ser una o más otras celdas o valores particulares denominados dentro de la expresión. La edición de usuario de la interfase de recálculo puede hacer que se vuelvan a ejecutar uno o más de las transformaciones, ocasionando así el recálculo.

La compilación involucra analizar la cadena de transformación de la interfase de usuario de recálculo para dependencias para crear una gráfica de dependencia de dependencias entre entidades. Por ejemplo, algunas dependencias pueden ser entre entidades con el fin de indicar que si una entidad es evaluada, entonces la otra también debe hacerlo. Otras dependencias pueden especificar eventos de usuario en los cuales depende la evaluación de una entidad. La gráfica de dependencia entonces se utiliza para crear un nivel inferior de pasos de ejecución. La gráfica de dependencia además se proporciona a un tiempo de ejecución para el programa, de manera que la gráfica de dependencia puede estar disponible durante la

operación de la interfase de usuario de recálculo.

Esta Breve Descripción no pretende identificar características clave o características esenciales de la materia objeto reclamada, no pretende ser utilizada como un auxiliar para determinar el alcance de la materia objeto reclamada.

BREVE DESCRIPCION DE LOS DIBUJOS

Con el fin de describir la manera en la cual las ventajas y características antes mencionadas y otras más pueden ser obtenidas, se presentará una descripción más particular de varias modalidades al hacer referencia a los dibujos anexos. Entendiendo que estos dibujos ilustran solo modalidades de muestra y, por lo tanto, no se consideran como limitantes del alcance de la invención, las modalidades serán descritas y explicadas con especificidad adicional y detalle a través del uso de los dibujos anexos en los cuales:

La Figura 1 de forma abstracta muestra un sistema de cómputo en donde se pueden emplear algunas modalidades aquí descritas;

La Figura 2 de forma abstracta muestra una interfase de usuario de recálculo ilustrativa, la cual muestra varias fuentes de datos y colectores de datos con transformaciones de intervención, y se utiliza como un ejemplo específico provisto para explicar los principios más amplio aquí descritos;

La Figura 3 muestra un ambiente de compilación ilustrativo que incluye un compilador que ingresa a la cadena de transformación y

produce un código compilado así como una cadena de dependencia;
y

La Figura 4 muestra un cuadro de flujo de un método para
compilar una cadena de transformación de una interfase de usuario
5 de recálculo;

La Figura 5 muestra un ambiente en donde los principios de la
presente invención pueden ser empleados, incluyendo una estructura
de composición dirigida por datos que construye una composición de
vista que depende de datos de entrada;

10 La Figura 6 muestra un ambiente de canalización que
representa un ejemplo del ambiente de la Figura 5;

La Figura 7 esquemáticamente muestra una modalidad de la
porción de datos de la canalización de la Figura 6;

La Figura 8 esquemáticamente muestra una modalidad de la
15 porción analítica de la canalización de la Figura 6; y

La Figura 9 esquemáticamente muestra una modalidad de la
porción de vista de la canalización de la Figura 6.

DESCRIPCION DETALLADA

20

Al menos algunas modalidades aquí descritas se refieren a la
compilación de una cadena de transformación de una interfase de
usuario de recálculo. La interfase de usuario de recálculo puede ser,
por ejemplo, un documento de recálculo, tal como una hoja de
25 cálculo. Sin embargo, la interfase de recálculo puede ser cualquier

lona electrónica presentada que incluye uno o más resultados de una cadena de transformación. La cadena de transformación incluye una pluralidad de transformaciones entre una fuente de datos respectiva y un colector de datos. La edición de usuario de la interfase de usuario de recálculo puede hacer que una o más de las transformaciones se vuelvan a ejecutar, ocasionando así el recálculo.

La compilación involucra el análisis de la cadena de transformación de la interfase de usuario de recálculo para dependencias para crear una gráfica de dependencia de dependencias entre entidades. Por ejemplo, algunas dependencias pueden ser entre entidades con el fin de indicar que si una entidad es evaluada, entonces la otra también debe ser. La gráfica de dependencia entonces se utiliza para crear un nivel inferior de pasos de ejecución. La gráfica de dependencia además es provista a un tiempo de ejecución para el programa, de manera que la gráfica de dependencia puede estar disponible durante la operación de la interfase de usuario de recálculo.

Con respecto a la Figura 1 se presentará alguna discusión introductoria de un sistema de cómputo. Entones, la compilación de la cadena de transformación de la interfase de usuario de recálculo será descrita con respecto a las figuras subsecuentes.

Los sistemas de cómputo ahora enormemente han tomado una amplia variedad de formas. Los sistemas de cómputo pueden ser, por ejemplo, dispositivos portátiles, aparatos, computadoras laptop,

computadoras de escritorio, ordenadores centrales, sistemas de cómputo distribuido, o aún dispositivos que convencionalmente no han sido considerados como un sistema de cómputo. En esta descripción y en las reivindicaciones, el término "sistema de 5 cómputo" se define ampliamente como incluyendo cualquier dispositivo o sistema (o una combinación de los mismos) que incluye al menos un procesador físico tangible, y una memoria física y tangible capaz de tener ahí instrucciones ejecutables por computadora que pueden ser ejecutadas por el procesador. La 10 memoria puede tomar cualquier forma y puede depender de la naturaleza y forma del sistema de cómputo. Un sistema de cómputo puede ser distribuido a través de un ambiente de red y puede incluir múltiples sistemas de cómputo constituyentes.

Como se muestra en la Figura 1, en su configuración más 15 básica, un sistema de cómputo 100 típicamente incluye al menos una unidad de procesamiento 102 y memoria 104. La memoria 104 puede ser una memoria de sistema física, la cual puede ser volátil, no volátil, o alguna combinación de las dos. El término "memoria" también se puede utilizar aquí para referirse a almacenamiento 20 masivo no volátil tal como medios de almacenamiento físicos. Si el sistema de cómputo es distribuido, el procesamiento, memoria y/o capacidad de almacenamiento también pueden ser distribuidos. Como se utiliza aquí, el término "módulo ejecutable" o "componente ejecutable" puede referirse a objetos de software enrutamientos, o 25 métodos que pueden ser ejecutados en el sistema de cómputo. Los

diferentes componentes, módulos, motores, y servicios aquí descritos pueden ser implementados como objetos o procedimientos que se ejecutan en el sistema de cómputo (por ejemplo, como secuencias separadas).

5 En la descripción que sigue, se describen modalidades haciendo referencia a actos que se realizan por uno o más sistemas de cómputo. Si dichos actos se implementan en software, uno o más procesadores del sistema de cómputo asociado que realiza el acto dirigen la operación del sistema de cómputo en respuesta a tener
10 instrucciones ejecutables por computadora. Por ejemplo, dichas instrucciones ejecutables por computadora pueden ser modalizadas en uno o más medios legibles por computadora que forman un producto de programa de computadora. Un ejemplo de dicha operación involucra la manipulación de datos. Las instrucciones
15 ejecutables por computadora (y los datos manipulados) pueden ser almacenadas en la memoria 104 del sistema de cómputo 100. El sistema de cómputo 100 también puede contener canales de comunicación 108 que permiten que el sistema de cómputo 100 se comunique con otros procesadores de mensaje a través de, por
20 ejemplo, la red 110. El sistema de cómputo 100 también incluye una presentación 112, la cual se puede utilizar para presentar representaciones visuales a un usuario.

Las modalidades aquí descritas pueden comprender o utilizar una computadora de propósito especial o de propósito general que
25 incluye hardware de computadora, tal como, por ejemplo, uno o más

procesadores y memoria de sistema, como se discute con mayor detalle más adelante. Las modalidades aquí descritas también incluyen medios legibles por computadora físicos y otros más para realizar o almacenar instrucciones ejecutables por computadora y/o

5 estructuras de datos. Dichos medios legibles por computadora pueden cualquier medio disponible que puede ser accedido por un sistema de computadora de propósito general o propósito especial. Los medios legibles por computadora que almacenan instrucciones ejecutables por computadora son medios de almacenamiento físicos.

10 Los medios legibles por computadora que pueden realizar instrucciones ejecutables por computadora son medios de transmisión. Así, manera de ejemplo, y no de limitación, las modalidades de la invención pueden comprender al menos dos tipos muy diferentes de medios legibles por computadora: medios de

15 almacenamiento de computadora y medios de transmisión.

Los medios de almacenamiento de computadora incluye RAM, ROM, EEPROM, CD-ROM, u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio tangible, el cual

20 puede ser utilizado para almacenar medios de código de programa deseados en la forma de instrucciones ejecutables por computadora o estructuras de datos y que pueden ser accedidos por una computadora de propósito general o de propósito especial.

Una ñredò se define como uno o más enlaces de datos que

25 permiten el transporte de datos electrónicos entre sistemas de

computadora y/o módulos y/u otros dispositivos electrónicos. Cuando la información es transferida o provista a través de una red u otra conexión de comunicaciones (ya sea por cable, inalámbrica, o una combinación de por cable o inalámbrica) a una computadora, la computadora apropiadamente ve la conexión como un medio de transmisión. Los medios de transmisión pueden incluir una red y/o enlaces de datos que se pueden utilizar para llevar o desear medios de código de programa en la forma de instrucciones ejecutables por computadora o estructuras de datos que pueden ser accedidas por la computadora de propósito general o propósito especial. Dentro del alcance de medios legibles por computadora también se deben incluir combinaciones de los anteriores.

Además, después de llegar a varios componentes de sistema de computadora, los medios de código de programa en la forma de instrucciones ejecutables por computadora o estructuras de datos pueden ser transferidos automáticamente desde medios de transmisión a medios de almacenamiento por computadora (o viceversa). Por ejemplo, las instrucciones ejecutables por computadora o estructuras de datos recibidas a través de una red o colector de datos pueden ser guardadas en memoria intermedia en RAM dentro de un módulo de interfase de red (por ejemplo, NIC), y luego finalmente transferirse a la RAM del sistema y/o a medios de almacenamiento de computadora menos volátiles en un sistema de computadora. De esta forma, se debe entender que los medios de almacenamiento de computadora pueden ser incluidos en

componentes de sistema de computadora que también (o aún principalmente) utilizan medios de transmisión.

Las instrucciones ejecutables por computadora comprenden, por ejemplo, instrucciones y datos que, cuando se ejecutan por un procesador, hacen que una computadora de propósito general, computadora de propósito especial, o dispositivo de procesamiento de propósito especial para realizar cierta función o grupo de funciones. Las instrucciones ejecutables por computadora pueden ser, por ejemplo, binarios, instrucciones de formato intermedio tales como lenguaje de ensamble, o aún código fuente. Aunque la materia objeto ha sido descrita en un lenguaje específico a características estructurales y/o actos metodológicos, se debe entender que la materia objeto definida en las reivindicaciones anexas no necesariamente está limitada a las características descritas o actos descritos anteriormente. Más bien las características y actos descritos se describen como formas ilustrativas para implementar las reivindicaciones.

Aquellos expertos en la técnica apreciarán que la invención puede ser practicada en ambientes de cómputo de red con muchos tipos de configuraciones de sistema de computadora, incluyendo, computadoras personales, computadoras de escritorio, computadoras laptop, procesadores de mensaje, dispositivos portátiles, sistemas de procesador múltiple, electrónica de consumidor a base de microprocesador o programable, PC de red, minicomputadoras, ordenadores centrales, teléfonos móviles, PDA, localizadores,

enrutadores, conmutadores, y similares. La invención también puede ser practicada en ambientes de sistema distribuido, en donde sistemas de computadora locales y remotos que están enlazados (ya sea a través de enlaces de datos por cable, enlaces de datos
5 inalámbricos, o a través de una combinación de enlaces de datos por cable o inalámbricos) a través de una red, ambos realizan tareas. En un ambiente de sistema distribuido, los módulos de programa pueden estar ubicados en dispositivos de almacenamiento de memoria tanto local como remota.

10 En esta descripción y en las reivindicaciones, una "interfase de usuario de recálculo" es una interfase con la cual un usuario puede interactuar y que ocurre en un ambiente en donde hay una o mas fuentes de datos y uno o más colectores de datos. Además, existe un grupo de transformaciones que cada una puede ser declarativamente
15 definida entre una o más fuentes de datos y un colector de datos. Por ejemplo, la salida de una fuente de datos es alimentada a la transformación, y el resultado de la transformación después es provisto al colector de datos, dando como resultado potencialmente algún tipo de cambio en visualización al usuario.

20 Las transformaciones son "declarativas" en el sentido de que un usuario, sin conocimiento de codificación específico, puede escribir las declaraciones que definen la transformación. Ya que la transformación es declarativamente definida, un usuario puede cambiar la transformación declarativa. En respuesta, se realiza el
25 recálculo, dando como resultado tal vez diferentes datos que son

provistos a los colectores de datos.

Un ejemplo clásico de una interfase de usuario de recálculo es un documento de hoja de cálculo. Un documento de hoja de cálculo incluye una rejilla de celdas. Inicialmente, las celdas están vacías, y de esta forma cualquier celda del programa de hoja de cálculo tiene el potencial de ser una fuente de datos o un colector de datos, dependiendo del significado y contexto de las expresiones declarativas ingresadas por un usuario. Por ejemplo, un usuario puede seleccionar una celda dada, y escribir una expresión en esa celda. La expresión puede ser tan simple como un valor escalar expresado que será asignado a esa celda. Esa celda después puede ser utilizada como una fuente de datos. Alternativamente, la expresión para una celda dada puede estar en la forma de una ecuación en donde los valores de entrada se toman de una o más de otras celdas. En ese caso, la celda dada es un colector de datos que presenta el resultado de la transformación. Sin embargo, durante una autoría continua, esa celda puede ser utilizada como un colector de datos para otras transformaciones declarativamente hechas por el autor.

El autor de un documento de hoja de cálculo no necesita ser un experto en código imperativo. El autor simplemente hace declaraciones que definen una transformación, y selecciona colectores de datos y fuentes de datos correspondientes. Las Figuras 5 a 9 descritas después proporcionan un ambiente de autoría declarativo más generalizado en donde se describe una interfase de

usuario de recálculo más generalizada. En ese ambiente subsecuentemente descrito, los controles visualizados pueden servir tanto como fuentes de datos como colectores de datos. Además, las transformaciones declarativas pueden ser más intuitivamente de más
5 autoría por simples manipulaciones de esos controles.

La Figura 2 de manera abstracta muestra una interfase de usuario de recálculo ilustrativa 200, la cual es un ejemplo específico provisto para explicar los principios más amplios aquí descritos. La interfase de usuario de recálculo 200 es solo un ejemplo de cómo los
10 principios aquí descritos pueden ser aplicados a cualquier interfase de usuario de recálculo para crear una variedad incontable de interfases de usuario de recálculo para una variedad incontable de aplicaciones.

La interfase de usuario de recálculo 200 incluye varias
15 transformaciones declarativas 211 a 215. El círculo punteado alrededor de cada una de las flechas que representan las transformaciones 211 a 216 simboliza que las transformaciones están cada una en forma declarativa.

En este ejemplo específico de la Figura 2, la transformación
20 211 incluye una fuente de datos 201 y un colector de datos 202 respectivos. Observar que un colector de datos para una transformación también puede ser una fuente de datos para otra transformación. Por ejemplo, el colector de datos 202 para la transformación 211 también sirve como una fuente de datos para la
25 transformación 212. Además, una transformación puede tener

múltiples fuentes de datos. De esta manera, la cadena de transformación puede ser jerárquica, y así bastante compleja. Por ejemplo, la transformación 212 incluye la fuente de datos 202 y el colector de datos 203. El colector de datos 203 incluye dos fuentes de datos; principalmente la fuente de datos 202 para la transformación 212, y la fuente de datos 205 para la transformación 214. Es decir, tal vez una sola transformación conduce las dos fuentes de datos 202 y 205 al colector de datos 203. La transformación 213 incluye una fuente de datos 204 y un colector de datos 205.

Si la interfase de usuario de recálculo fuera un documento de hoja de cálculo, por ejemplo, las varias fuentes/colectores de datos 201 a 205 pueden ser celdas de hoja de cálculo, en tal caso las transformaciones representan la expresión que puede ser asociada con cada colector de celda. La salida de cada expresión es presentada dentro de la celda. De esta manera, en el caso de una hoja de cálculo, las fuentes/colectores de datos pueden ser controles visualizados complejos que ambas incluyen parámetros de entrada a y parámetros de salida de la cadena de transformación. Por ejemplo, en la Figura 2, hay una transformación declarativa adicional 215 que va de la fuente de datos 205 al colector de datos 201. Así, la fuente/colector de datos 201 puede visualizar información que representa una salida desde la transformación 215, así como proporcionar datos adicionales a otros colectores de datos.

Las interfases de usuario de recálculo no necesitan tener

controles de visualización. Un ejemplo de esto es una interfase de usuario de recálculo hecha para realizar una computación basada en transformación, consumir datos de fuente y actualizar datos de colector, sin ninguna información presentada al usuario sobre la computación en el caso normal. Por ejemplo, la interfase de usuario de recálculo puede soportar una computación de fondo. Un segundo ejemplo es una interfase de usuario de recálculo que tiene controles de salida que operan accionadores externos, tales como las válvulas en el control de procedimiento ilustrativo. Dichos controles son como controles de presentación ya que sus estados son controlados por resultados de la computación de transformación y de entradas de señal. Sin embargo, aquí, la salida es una señal de control a un dispositivo en lugar de una visualización a una presentación. Considerar, por ejemplo, una interfase de usuario de recálculo para controlar un robot. Esta interfase de usuario de recálculo puede tener reglas para acciones y comportamiento del robot que dependen de sensores de robot de entrada como servo-posiciones y velocidades, mediciones de hallazgo de escala ultrasónica, y así sucesivamente. O considerar una aplicación de control de procedimiento basada en una interfase de usuario de recálculo que toma señales de sensores de equipo como posiciones de válvula, velocidades de flujo de fluido, etc.

La Figura 3 muestra un ambiente de compilación 300 ilustrativo que incluye un compilador 310 que tiene acceso a la cadena de transformación 301. Un ejemplo de la cadena de transformación 301

es la cadena de transformación 200 de la Figura 2. La Figura 4 muestra un cuadro de flujo de un método 400 para compilar una cadena de transformación de una interfase de usuario de recálculo. El método 400 puede ser realizado por el compilador 310 de la
5 Figura 3. En una modalidad, el método 400 puede ser realizado por el sistema de cómputo 100 en respuesta al procesador(es) 102 que ejecuta instrucciones ejecutables por computadora modalizadas en uno o más medios de almacenamiento legibles por computadora.

El método 400 incluye analizar una cadena de transformación
10 de la interfase de usuario de recálculo para dependencias (acto 401). Por ejemplo, haciendo referencia a la Figura 2, el compilador 300 puede analizar cada una de las transformaciones 211 a 215. Las transformaciones son declarativas y de esta forma las dependencias pueden ser extraídas más fácilmente que lo que podría ser si las
15 transformaciones se expresaran utilizando un lenguaje de computadora imperativo.

Basándose en el análisis, se crea una gráfica de dependencia (acto 402) entre entidades denominadas en las transformaciones. Esencialmente, las dependencias tienen una entidad de fuente que
20 representa un evento, y una entidad objetivo que representa que la evaluación de esa entidad objetivo depende del evento. Un ejemplo del evento podría ser un evento de usuario en donde el usuario interactúa, de cierta forma, con la interfase de usuario de recálculo. Como otro ejemplo, el evento puede ser un evento de inter-entidad
25 en donde si se evalúa la entidad de fuente, entonces la entidad

objetivo de la dependencia también debe ser evaluada.

El compilador luego crea pasos de ejecución de nivel inferior basándose en la gráfica de dependencia (acto 403). Los pasos de ejecución de nivel inferior, por ejemplo, pueden ser de un código de lenguaje imperativo. El código de lenguaje imperativo está adaptado para responder a eventos de detección, referencia a un cuadro de evento para determinar una función a ejecutar, y ejecutar esa función. Por consiguiente, cada una de las dependencias en la gráfica de dependencia puede ser reducida a una función. La gráfica de dependencia por si misma puede ser provista al tiempo de ejecución (acto 404). El código de lenguaje imperativo puede ser, por ejemplo, un lenguaje de guión o script, tal como JAVASCRIPT. Sin embargo, los principios aquí descritos no están limitados a que el código de lenguaje imperativo sea de cualquier lenguaje particular.

Como un ejemplo, la Figura 3 muestra que el compilador genera también el código de nivel inferior. Dicho código de nivel inferior incluye una compilación de cada una de las transformaciones en la cadena de transformación. Por ejemplo, el código de nivel inferior se muestra como incluyendo un elemento que representa la compilación de cada una de las transformaciones en la cadena de transformación. En el contexto de la Figura 2, el elemento podría incluir una compilación de cada una de las transformaciones 211 a 215. El código de nivel inferior también incluye una variedad de funciones. Una función se genera para cada dependencia en la gráfica de dependencia. Las

funciones pueden ser funciones de lenguaje imperativo.

Cuando un tiempo de ejecución de lenguaje imperativo detecta un evento que está listado en la gráfica de dependencia, la función correspondiente dentro de las funciones compiladas 322 también se ejecuta. Por consiguiente, con todas las transformaciones apropiadamente compiladas, y con cada una de las dependencias en eventos particulares impuestas por las funciones dedicadas, la interfase de usuario de recálculo declarativa es apropiadamente representada como un código de lenguaje imperativo.

Así, se ha descrito un mecanismo efectivo para compilar una interfase de usuario de recálculo declarativa. Además, el tiempo de ejecución es provisto con una gráfica de dependencia, en lugar de un intérprete más extensivo.

Con respecto a las Figuras 5 a 9, ahora se describirá un ejemplo específico de una canalización de autoría para permitir que no programadores escriban programas con comportamientos complejos utilizando una interfase de usuario de recálculo.

La Figura 5 muestra un ambiente de composición visual 500 que se puede utilizar para construir una composición visual interactiva en la forma de una interfase de usuario de recálculo. La construcción de la interfase de usuario de recálculo se realiza utilizando analítica dirigida por datos y visualización de los resultados analíticos. El ambiente 500 incluye una estructura de composición 510 que realiza lógica que se hace independiente del dominio de problema de la composición de vista 530. Por ejemplo, la

misma estructura de composición 510 puede ser utilizada para hacer composiciones de vista interactiva para planos de ciudad, módelos moleculares, diseños de estantes de alimentos, desempeño de máquina o análisis de ensamble, u otras presentaciones de dominio
5 específico.

La estructura de composición 510 utiliza datos de dominio específico 520, sin embargo, para construir la composición visual actual 530 que es específica al dominio. Por consiguiente, la misma estructura de composición 510 puede ser utilizada para interfases de
10 usuario de recálculo para cualquier número de diferentes dominios al cambiar los datos de dominio específico 520, en lugar de tener que re-codificar la misma estructura de composición 510. De esta forma, la estructura de composición 510 de la canalización 500 puede aplicarse a un número potencialmente ilimitado de dominios de
15 problema, o al menos a una amplia variedad de dominios de problema, al alterar datos, en lugar de re-codificar y re-compilar. La composición de vista 530 luego puede ser suministrada como instrucciones a un módulo de presentación 2-D o 3-D apropiado. La arquitectura aquí descrita también permite la incorporación
20 conveniente de modelos de composición de vista pre-existentes como bloques de construcción para nuevos modelos de composición de vista. En una modalidad, se pueden incluir múltiples composiciones de vista en una composición de vista integrada para permitir una fácil comparación entre dos posibles soluciones a un modelo.

25 La Figura 6 muestra una arquitectura ilustrativa de la

estructura de composición 510 en la forma de un ambiente de canalización 600. El ambiente de canalización 600 incluye, entre otras cosas, la misma canalización 601. La canalización 601 incluye una porción de datos 610, una porción analítica 620, y una porción de vista 630, cada de las cuales se describirá con detalle con respecto a las Figuras 7 a 9 subsecuentes, respectivamente, y la descripción acompañante. Hora, a un nivel general, la porción de datos 610 de la canalización 610 puede aceptar una variedad de diferentes tipos de datos y presenta esos datos en una forma canónica a la porción analítica 620 de la canalización 601. La porción analítica 620 une los varios parámetros de modelo, y resuelve lo desconocido en los parámetros de modelo utilizan analítica de modelo. Los varios valores de parámetro después se proporcionan a la porción de vista 630, la cual construye la vista compuesta utilizando esos valores de los parámetros de modelo.

El ambiente de canalización 600 también incluye un componente de autoría 640 que permite a un autor u otro usuario de la canalización 601 formular y/o seleccionar datos para proporcionar a la canalización 601. Por ejemplo, el componente de autoría 640 puede ser utilizado para suministrar datos a cada una de la porción de datos 610 (representada por datos de entrada 611), porción analítica 620 (representada por datos analíticos 621), y porción de vista 630 (representada por datos de vista 631). Los varios datos 611, 621 y 631 representan un ejemplo de los datos de dominio específico 520 de la Figura 5, y se describirán con mayor detalle más

adelante. El componente de autoría 640 apoya la provisión de una amplia variedad de datos incluyendo, por ejemplo, esquemas de datos, datos actuales que se utilizarán por el modelo, la ubicación o escala de ubicaciones posible de datos que se van a traer de fuentes
5 externas, objetos visuales (gráficos o de animación), interacciones de interfase de usuario que se pueden realizar en declaraciones visuales, de modelación (por ejemplo, vistas, ecuaciones, restricciones), uniones, etc. En una modalidad, el componente de autoría es solo una porción de la funcionalidad provista por un
10 componente administrador completo (no mostrado en la Figura 6, pero representado por la estructura de composición 510 de la Figura 5). El administrador es un director total que controla y da secuencia a la operación de todos los otros componentes (tales como conectores de datos, solucionadores, visores, etc.), en respuesta a
15 eventos (tales como eventos de interacción de usuario, eventos de datos externos, y eventos de cualquiera de los otros componentes tales como los solucionadores, el sistema operativo, etc.).

En el ambiente de canalización 600 de la Figura 6, el componente de autoría 640 se utiliza para proporcionar datos a una
20 canalización existente 601, en donde están los datos que dirigen todo el procedimiento desde definir los datos de entrada, definir el modelo analítico (denominado anteriormente como la ñcadena de transformaciónñ), para definir cómo los resultados de la cadena de transformación se visualizan en la composición de vista. Por
25 consiguiente, no se necesita realizar ninguna codificación con el fin

de adaptar la canalización 601 a cualquiera de una de una variedad de dominios y problemas. Solo los datos provistos a la canalización 601 es lo que va a cambiar con el fin de aplicar la canalización 601 para visualizar una composición de vista diferente ya sea de un
5 diferente dominio de problema, o tal vez ajustar el problema que resuelve un dominio existente. Además, ya que los datos pueden ser cambiado en un tiempo de uso (es decir, tiempo de ejecución), así como el tiempo de autor, el modelo puede ser modificado y/o extendido en el tiempo de ejecución. De esta manera, hay menos
10 distinción, si la hay, entre la creación de un modelo y la ejecución del modelo. Ya que la autoría involucra editar artículos de datos y ya que el software corre todo su comportamiento de datos, cada cambio a los datos inmediatamente afecta el comportamiento sin necesidad de registrar y recompilar.

15 El ambiente de canalización 600 también incluye un módulo de respuesta de interacción de usuario 650 que detecta cuando un usuario ha interactuado con la composición de vista presentada, y después determina que hacer en respuesta. Por ejemplo, algunos tipos de interacciones pueden requerir de ningún cambio en los datos
20 provistos a la canalización 601 y así no se requiere de ningún cambio a la composición de vista. Otros tipos de interacciones pueden cambiar uno o más de los datos 611, 621, o 631. En ese caso, estos datos nuevos o modificados pueden hacer que nuevos datos sean provistos a la porción de datos 610, pueden requerir de
25 un re-análisis de los datos de entrada por la porción analítica 620,

y/o pueden requerir de re-visualización de la composición de vista por la porción de vista 630.

Por consiguiente, la canalización 601 se puede utilizar para extender visualizaciones analíticas dirigidas por datos tal vez a un número ilimitado a de dominios de problema, o al menos a una amplia variedad de dominios de problema. Además, no se necesita ser un programador para alterar la composición de vista para dirigir una amplia variedad de problemas. Cada una de la porción de datos 610, la porción analítica 620 y la porción de vista 630 de la canalización 601 ahora será descrita con respecto a la porción de datos 700 respectiva de la Figura 7, la porción analítica 800 de la Figura 8, y la porción de vista 900 de la Figura 9, en ese orden. Como será evidente a partir de las Figuras 7 a 9, la canalización 601 puede ser construida como una serie de componentes de transformación, en donde cada uno 1) recibe algunos datos de entrada apropiados, 2) realiza alguna acción en respuesta a esos datos de entrada (tal como realizar una transformación en los datos de entrada), y 3) datos de salida que después sirven como datos de entrada al siguiente componente de transformación.

La Figura 7 muestra solo una de muchas posibles modalidades de la porción de datos 700 de la canalización 601 de la Figura 6. Una de las funciones de la porción de datos 700 es proporcionar datos en un formato canónico que es consistente con esquemas entendidos por la porción analítica 800 de la canalización discutida con respecto a la Figura 8. La porción de datos incluye un componente de acceso

de datos 710 que accede a datos heterogénicos 701. Los datos de entrada 701 pueden ser ñheterogénicosò en el sentido de que los datos pueden (pero no necesitan) ser presentados al componente de acceso de datos 710 en una forma canónica. En realidad, la porción

5 de datos 700 se estructura de manera que los datos heterogénicos pueden ser de una amplia variedad de formatos. Ejemplos de diferentes tipos de datos de dominio que pueden ser accedidos y operados por modelos incluyen documentos de texto, y XML, tablas, listas, jerarquías (árboles), resultados de consulta de base de datos

10 SQL, resultados de consulta de cubo de inteligencia de negocios (BI), información gráfica tal como dibujos 2D y modelos visuales 3D en varios formatos, y sus combinaciones (es decir, un un compuesto). Además, el tipo de datos que pueden ser accedidos puede ser extendido declarativamente, al proporcionar una definición

15 (por ejemplo, un esquema) para los datos que serán accedidos. Por consiguiente, la porción de datos 700 permite una amplia variedad de entrada heterogénica en el modelo. Y también soporta el tiempo de ejecución, extensión declarativa de tipos de datos accesibles.

En una modalidad, la porción de acceso a datos 700 incluye un

20 número de conectores para obtener datos de un número de diferentes fuentes de datos. Ya que una de las funciones principales del conector es colocar datos correspondientes en forma canónica, dichos conectores por lo general serán denominados de aquí en adelante y en los dibujos como ñcanonicalizadoresò. Cada

25 canonicalizador puede tener un entendimiento de las Interfases de

de Programa de Aplicación específicas (APIs) de su fuente de datos correspondientes. El canonicalizador también puede incluir la lógica correspondiente para interconectar con esa API correspondiente para leer y/o escribir datos de y a la fuente de datos. De esta forma, los
5 canonicalizadores unen fuentes de datos externas y la imagen de memoria de los datos.

El componente de acceso a datos 710 evalúa los datos de entrada 701. Si los datos de entrada ya son canónicos y de esta forma procesables por la porción analítica 800, entonces los datos
10 de entrada pueden ser directamente provistos como datos canónicos 740 para ser ingresados a la porción analítica 800.

Sin embargo, si los datos de entrada 701 no son canónicos, entonces el componente de canonicalización de datos apropiado 730 es capaz de convertir los datos de entrada 701 al formato canónico.
15 Los componentes de canonicalización de datos 730 en realidad son una colección de componentes de canonicalización 730, cada uno capaz de convertir los datos de entrada que tienen características particulares en forma canónica. La colección de componentes de canonicalización 730 se ilustra como incluyendo cuatro componentes
20 de canonicalización 731, 732, 733 y 734. Sin embargo, las elipses 735 representan que puede haber cualquier otro número de componentes de canonicalización también, tal vez aún menos que los cuatro ilustrados.

Los datos de entrada 701 aún pueden incluir un mismo
25 canonicalizador así como un identificador de características de datos

correlacionadas. La porción de datos 700 entonces puede registrar las características de datos correlacionadas, y proporcionar el componente de canonicalización a la colección de componente de canonicalización de datos 730, en donde puede ser agregado a los componentes de canonicalización disponibles. Si se reciben después los datos de entrada que aquellas características correlacionadas, la porción de datos 710 entonces puede asignar los datos de entrada al componente de canonicalización correlacionada. Los componentes de canonicalización también se pueden encontrar dinámicamente de fuentes externas, tales como de bibliotecas de componente definidas en la web. Por ejemplo, si se conoce el esquema de una fuente de datos dada pero el canonicalizador no está presente, el canonicalizador puede ser localizado de una biblioteca de componente externa, siempre que una biblioteca puede ser encontrada y contenga los componentes necesarios. La canalización también puede analizar datos para los cuales ningún esquema aún es conocido y comparar los resultados de análisis contra la información de esquema en bibliotecas de componente conocidas para intentar una determinación dinámica del tipo de los datos, y así localizar los componentes de canonicalizador necesarios.

Alternativamente, en lugar de que los datos de entrada incluyan todo lo del componente de canonicalización, los datos de entrada más bien proporcionar una definición de transformación que define transformaciones de canonicalización. La colección 730 entonces puede ser configurada para convertir esa definición de

transformaciones en un componente de canonicalización correspondientes que impone las transformaciones junto con cero o más transformaciones de canonicalización por omisión estándares. Esto representa un ejemplo de un caso en donde la porción de datos
5 700 consume los datos de entrada y no proporciona datos canonicalizados correspondientes por abajo de la canalización. Sin embargo, tal vez en la mayoría de los casos, los datos de entrada 701 dan como resultado la generación de datos canonicalizados 740 correspondientes.

10 En una modalidad, la porción de datos 710 puede ser configurada para asignar datos de entrada al componente de canonicalización de datos con base en un tipo de archivo y/o tipo de formato de los datos de entrada. Otras características pueden incluir, por ejemplo, una fuente de los datos de entrada. Un componente de
15 canonicalización por omisión puede ser asignado a datos de entrada que no tienen un componente de canonicalización correspondiente designado. El componente de canonicalización por omisión puede aplicar un conjunto de reglas para intentar canonicalizar los datos de entrada. Si el componente de canonicalización por omisión no es
20 capaz de canonicalizar los datos, el componente de canonicalización por omisión puede accionar el componente de autoría 540 de la Figura 5 para incitar al usuario a proporcionar una definición de esquema para los datos de entrada. Si una definición de esquema todavía no existe, el componente de autoría 540 puede presentar un
25 asistente de definición de esquema para ayudar al autor a generar

una definición de esquema correspondiente que se puede utilizar para transformar los datos de entrada en forma canónica. Una vez que los datos están en forma canónica, el esquema que acompaña a los datos proporciona una descripción suficiente de los datos que el
5 resto de la canalización 601 no necesita un nuevo código para interpretar los datos. Más bien, la canalización 601 incluye un código que es capaz de interpretar datos en vista de cualquier esquema que se puede expresar en un lenguaje de declaración de esquema accesible.

10 Independientemente, los datos canónicos 740 se proporcionan como datos de salida de la porción de datos 700 y como datos de entrada a la porción analítica 800. Los datos canónicos pueden incluir campos que incluyen una variedad de tipos de datos. Por ejemplo, los campos pueden incluir tipos de datos simples tales como
15 enteros, números de punto flotantes, secuencias, vectores, disposiciones, colecciones, estructuras jerárquicas, texto, documentos XML, tablas, listas, resultados de consulta de base de datos SQL, resultados de consulta de cubo de inteligencia de negocios (BI), información gráfica tal como dibujos 2D y modelos
20 visuales 3D en varios formatos, o aún combinaciones complejas de estos varios tipos de datos. Como otra ventaja, el procedimiento de canonicalización es capaz de canonicalizar una amplia variedad de datos de entrada. Además, la variedad de datos de entrada que la porción de datos 700 es capaz de aceptar es expandible. Esto es de
25 gran ayuda en el caso en donde se combinan múltiples modelos,

como se discutirá posteriormente en esta descripción.

La Figura 8 muestra la porción analítica 800 que representa un ejemplo de la porción analítica 620 de la canalización 601 de la Figura 6. La porción de datos 700 proporciona los datos
5 canonicalizados 801 al componente de unión de modelo de datos 810. Aunque los datos canonicalizados 801 pueden tener cualquier forma canonicalizada, y cualquier número de parámetros, en donde la forma y el número de parámetros aún puede diferir de una pieza de datos de entrada a otra. Para propósitos de discusión, sin embargo,
10 los datos canónicos 801 tienen campos 802A al 802H, los cuales colectivamente pueden ser denominados aquí como "campos 802".

Por otro lado, la porción analítica 800 incluye un número de parámetros de modelo 811. El tipo y el número de parámetros de modelo pueden diferir de acuerdo con el modelo. Sin embargo, para
15 propósitos de discusión de un ejemplo particular, los parámetros de modelo 811 serán discutidos como incluyendo parámetros de modelo 811A, 811B, 811C y 811D. En una modalidad, la identidad de los parámetros de modelo, y las relaciones analíticas entre los parámetros de modelo pueden ser declarativamente definidos sin
20 utilizar codificación imperativa.

Un componente de unión de modelo de datos 810 intercede entre los campos de datos canonicalizados 802 y los parámetros de modelo 811 para proporcionar así uniones entre los campos. En este caso, el campo de datos 802B está unido al parámetro de modelo
25 811A como se representa por la flecha 803A. En otras palabras, el

valor del campo de datos 802B se utiliza para llenar el parámetro de modelo 811A. También, en este ejemplo, el campo de datos 802E está unido al parámetro de modelo 811B (según representado por la flecha 803B), y el campo de datos 802H está unido al parámetro de modelo 811C (según representado por la flecha 803C).

Los campos de datos 802A, 802C, 802D, 802F y 802G no se muestran unidos a ninguno de los parámetros de modelo. Esto es para enfatizar que no todos los campos de datos de los datos de entrada son siempre requeridos para ser utilizados como parámetros de modelo. En una modalidad, uno o más de estos campos de datos pueden ser utilizados para proporcionar instrucciones al componente de unión de modelo de datos 810 en donde los campos de los datos canonicalizados (para estos datos canonicalizados o tal vez cualesquiera datos canonicalizados futuros) se van a unir a dicho parámetro de modelo. Esto representa un ejemplo del tipo de datos analíticos 621 que pueden ser provistos a la porción analítica 620 de la Figura 6. La definición de qué campos de datos de los datos canonicalizados se unen a los parámetros de modelo pueden ser formulada en un número de formas. Por ejemplo, las uniones pueden ser 1) explícitamente establecidas por el autor en un tiempo de autoría, 2) explícitamente establecidas por el usuario como tiempo de uso (sometido a cualquier restricción impuesta por el autor), 3) automáticamente unidas por el componente de autoría 640 basándose en heurística algorítmica, y/o 4) incitadas por el componente de autoría del autor y/o usuario para especificar una

unión cuando se determina que una unión no puede hacerse algorítmicamente. De esta manera, las uniones también pueden ser resueltas como parte de la misma lógica de modelo.

La habilidad de un autor para definir cual de los campos de
5 datos se traza a qué parámetros de modelo proporcionar al autor una gran flexibilidad de ser capaz de utilizar símbolos de que el autor está cómodo para definir parámetros de modelo. Por ejemplo, si uno de los parámetros de modelo representa presión, el autor puede nombrar a ese parámetro de modelo como "Presión" o "P" o cualquier
10 otro símbolo que hace sentido al autor. El autor aún puede renombrar el parámetro de modelo que, en una modalidad, puede hacer que el componente de unión de modelo 810 automáticamente actualice para permitir uniones que previamente fueron para el parámetro de modelo del nombre antiguo en lugar de unirse al parámetro de
15 modelo del nuevo nombre, conservando así las uniones deseadas. Este mecanismo para unir también puede permitir que la unión sea cambiada declarativamente en el tiempo de ejecución.

El parámetro de modelo 811D se ilustra con un asterisco para enfatizar que en este ejemplo, al parámetro de modelo 811D no se le
20 asignó un valor por el componente de unión de modelo de datos 810. Por consiguiente, el parámetro de modelo 811D permanece desconocido. En otras palabras, al parámetro de modelo 811D no se le asigna un valor.

El componente de modelación 820 realiza un número de
25 funciones. Primero, el componente de modelación 820 define

relaciones analíticas 821 entre los parámetros de modelo 811. Las relaciones analíticas 821 están categorizadas en tres categorías generales incluyendo ecuaciones 831, reglas 832 y restricciones 833. Sin embargo, la lista de solucionadores es extensible. En una
 5 modalidad, por ejemplo, se pueden incorporar una o más simulaciones como parte de las relaciones analíticas provistas, se proporcionar un motor de simulación correspondiente y se registra como un solucionador.

El término "ecuación", como se utiliza aquí, se alinea con el
 10 término que se utiliza en el campo de matemáticas.

El término "reglas", como se utiliza aquí, significa una declaración condicional en donde si una o más condiciones se satisfacen (la condicional o porción "si" de la declaración condicional), entonces se tomarán una o más acciones (la
 15 consecuencia o porción "entonces" de la declaración condicional). Una regla es aplicada a los parámetros de modelo si uno o más parámetros de modelo se expresan en la declaración condicional, o uno o más parámetros de modelo se expresan en la declaración de consecuencia.

20 El término "restricción", como se utiliza aquí, significa que se aplica una restricción a uno o más parámetros de modelo. Por ejemplo, en un modelo de planeación de ciudad, un elemento de casa particular puede ser restringido a colocarse en una ubicación de mapa que tiene un subgrupo de las designaciones de zona totales
 25 posibles. Un elemento de puente puede ser restringido por abajo de

cierta longitud máxima, o cierto número de carriles.

Un autor que está familiarizado con el modelo puede proporcionar expresiones de estas ecuaciones, reglas y restricciones que se aplican a ese modelo. En el caso de simulaciones, el autor
 5 puede proporcionar un motor de simulación apropiado que provee las relaciones de simulación apropiadas entre los parámetros de modelo. El componente de modelación 820 puede proporcionar un mecanismo para que el autor proporcione una expresión simbólica natural para las ecuaciones, reglas y restricciones. Por ejemplo, un autor del
 10 modelo relacionado con termodinámica simplemente puede copiar y pegar ecuaciones de un libro de texto de termodinámica. La habilidad de unir parámetros de modelo a campos de datos permite que el autor utilice cualquier símbolo con el que el autor sea familiar (tal como los símbolos exactos utilizados en libros de texto basados en
 15 autor) o los símbolos exactos que al autor le gusta usar.

Antes de la resolución, el componente de modelación 820 también identifica cual de los parámetros de modelo se van a resolver (es decir, de aquí en adelante, la ñvariable de modelo de salidaò si es singular, o ñvariables de modelo de salidaò si el plural,
 20 o ñvariable(s) de modelo de salidaò si es una variable de modelo de salida individual o variables de modelo de salida plurales). Las variables de modelo de salida pueden ser parámetros desconocidos, o pueden ser parámetros de modelo conocidos, en donde el valor del parámetro de modelo conocido es sometido a cambio en la operación
 25 de resolución. En el ejemplo de la Figura 8, después de la operación

de unión de modelo de datos. Los parámetros de modelo 811A, 811B y 811C son conocidos, el el parámetro de modelo 811D es desconocido. Por consiguiente, el parámetro de modelo 811D desconocido puede ser una de las variables de modelo de salida.

5 Alternativamente o además, uno o más de los parámetros de modelo conocidos 811A, 811B y 811C también pueden ser variables de modelo de salida. El solucionador 840 entonces resuelve la variable(s) de modelo de salida, si es posible. En una modalidad descrita más adelante, el solucionador 840 es capaz de resolver una

10 variedad de variables de modelo de salida, aún dentro de un modelo individual siempre que se proporcionen suficientes variables de modelo de entrada para permitir que se realice la operación de resolución. Las variables de modelo de entrada, por ejemplo, pueden ser parámetros de modelo conocidos cuyos valores no son sometidos

15 a cambio durante la operación de resolución. Por ejemplo, en la Figura 8, si los parámetros de modelo 811A y 811D son variables de modelo de entrada, el solucionador más bien puede resolverlas variables de modelo de salida 811B y 811C. En una modalidad, el solucionador puede dar salida a cualquiera de un número de

20 diferentes tipos de datos para un parámetro de modelo individual. Por ejemplo, algunas operaciones de ecuación (tales como adición, substracción, y similares) se aplican sin consideran si los operandos son enteros, punto de flotación, vectores de los mismos, o matrices de los mismos.

25 En una modalidad, aún cuando el solucionador 840 no puede

resolver una variable de modelo de salida particular, el solucionador 800 aún puede presentar una solución parcial para esa variable de modelo de salida, aún si una resolución completa al resultado numérico real (o cualquier cosa resuelta para el tipo de datos) no es posible. Esto permite que la canalización facilite un desarrollo en incremento al incitar al autor a qué información se necesita para llegar a una solución completa. Esto también ayuda a eliminar la distinción entre el tiempo de autor y el tiempo de uso, ya que al menos una solución parcial está disponible a través de las varias etapas de autoría. Para un ejemplo abstracto, suponer que el modelo analítico incluye una ecuación $a=b+c+d$. Ahora suponer que a , c y d son variables de modelo de salida, y b es una variable de modelo de entrada teniendo un valor conocido de 5 (un entero en este caso). En el procedimiento de solución, el solucionador 840 solo es capaz de resolver una de las variables de modelo de salida a , y asigna un valor de 6 (un entero) al parámetro de modelo denominado a , pero el solucionador 840 no es capaz de resolver para c . Ya que a depende de c , el parámetro de modelo denominado a también permanece como desconocido y sin resolver. En este caso, en lugar de asignar un valor entero a a , el solucionador puede hacer una solución parcial y producir el valor de secuencia de $c+11$ al parámetro de modelo a . Como se mencionó previamente, esto puede ser especialmente útil cuando un experto de dominio está creando un modelo analítico, y servirá esencialmente para proporcionar información parcial con respecto al contenido del

parámetro de modelo 810 y también servirá para decirle al autor que algo de analítica de modelo adicional necesita ser provista para que permita que el parámetro de modelo 810 sea resuelto. Este resultado de resolución parcial tal vez puede salir en alguna forma en la composición de vista para permitir que el experto de dominio vea el resultado parcial.

El solucionador 840 se muestra en una forma simplificada en la Figura 8. Sin embargo, el solucionador 840 puede dirigir la operación de múltiples solucionadores constituyentes como se describirá con respecto a la Figura 9. En la Figura 8, el componente de modelación 820 entonces hace a los parámetros de modelo (incluyendo las variables de modelo ahora conocidos y resueltos para salida) disponibles como la salida que será provista a la porción de vista 900 de la Figura 9.

La Figura 9 muestra una porción de vista 900, la cual representa un ejemplo de la porción de vista 630 de la Figura 6, y representa un ejemplo de controles visualizados en la interfase de usuario de recálculo 200. La porción de vista 900 recibe los parámetros de modelo 811 de la porción analítica 800 de la Figura 8. La porción de vista también incluye un depósito de componentes de vista 920 que contiene una colección de componentes de vista. Por ejemplo, el depósito de componentes de vista 920, en este ejemplo, se ilustra como incluyendo componentes de vista 921 al 924, aunque el depósito de componentes de vista 920 puede contener cualquier número de componentes. Los componentes de vista cada uno

incluyen de cero a más parámetros de entrada. Por ejemplo, el componente de vista 921 no incluye ningún parámetro de entrada. Sin embargo, el componente de vista 922 incluye dos parámetros de entrada 942A y 942B. El componente de vista 923 incluye un parámetro de entrada 943, y el componente de visión 924 incluye un parámetro de entrada 944. Es decir, esto solo es un ejemplo. Los parámetros de entrada pueden, pero no necesariamente, afectar cómo se presenta el artículo visual. El hecho de que el componente de vista 921 no incluye cualquier parámetro de entrada enfatiza que pueden ser vistas que se generen sin referencia a cualquier parámetro de modelo. Considerar una vista que comprende solo datos fijos (incorporados) que no cambian. Dicha vista, por ejemplo, puede constituir información de referencia para el usuario. Alternativamente, considerar una vista que solo proporciona una forma para navegar en un catálogo, de manera que se pueden seleccionar artículos del mismo para importación a un modelo.

Cada componente de vista 921 a 924 incluye o está asociado con lógica correspondiente que, cuando se ejecuta por el componente de composición de vista 940 utilizando el parámetro(s) de entrada de componente de vista correspondiente, si hay, hace que un artículo de vista correspondiente sea colocado en el espacio virtual 950. Ese artículo virtual puede ser una imagen u objeto estático, o puede ser un artículo u objeto virtual animado, dinámico. Por ejemplo, cada componente de vista 921 a 924 está asociado con la lógica 931 a 934 correspondiente que, cuando se ejecuta, hace que el artículo virtual

951 a 954 correspondiente, respectivamente, sea presentado en el espacio virtual 950. Los artículos virtuales se ilustran como formas simples. Sin embargo, los artículos virtuales pueden ser bastante complejos en forma tal vez aún incluyendo animación. En esta descripción, cuando un artículo de vista es presentado en un espacio virtual, esto significa que el componente de composición de vista ha creado suficientes instrucciones que, cuando se proporcionan al motor de presentación, el motor de presentación es capaz de presentar el artículo de vista en la presentación en la ubicación designada y en la forma designada.

Los componentes de vista 921 a 924 pueden ser provistos tal vez aún como datos de vista a la porción de vista 900 utilizando, por ejemplo, el componente de autoría 640 de la Figura 6. Por ejemplo, el componente de autoría 640 puede proporcionar un selector que permite al autor seleccionar de varias formas geométricas, o tal vez componer otras formas geométricas. El autor también puede especificar los tipos de parámetros de entrada para cada componente de vista, mientras que algunos de los parámetros de entrada pueden ser parámetros de entrada por omisión impuestos por la porción de vista 900. La lógica que está asociada con cada componente de vista 921 a 924 también puede proporcionar datos de vista, y/o también puede incluir alguna funcionalidad por omisión provista por la misma porción de vista 900.

La porción de vista 900 incluye un componente de unión de vista de modelo 910 que está configurado para unir al menos algunos

de los parámetros de modelo a parámetros de entrada correspondientes de los componentes de vista 921 a 924. Por ejemplo, el parámetro de modelo 811A está unido al parámetro de entrada 942A del componente de vista 922 como se representa por la flecha 911A. El parámetro de modelo 811B está unido al parámetro de entrada 942B del componente de vista 922 como se representa por la flecha 911B. También, el parámetro de modelo 811D está unido a los parámetros de entrada 943 y 944 de los componentes de visión 923 y 924, respetivamente, como se representa por la flecha 911C. El parámetro de modelo 811C no se muestra unido a ningún parámetro de componente de vista correspondiente, enfatizando que no todos los parámetros de modelo necesitan ser utilizados por la porción de vista de la canalización, aún si esos parámetros de modelo fueran esenciales en la porción analítica. También, el parámetro de modelo 811D se muestra unido a dos diferentes parámetros de entrada de componentes de vista representando que los parámetros de modelo pueden ser unidos a múltiples parámetros de componente de vista. En una modalidad, la definición de la uniones entre los parámetros de modelo y los parámetros de componente de vista pueden ser formuladas al 1) establecerse explícitamente por el autor en el tiempo de autoría, 2) establecerse explícitamente por el usuario en el tiempo de uso (sometido a cualquier restricción impuesta por el autor), 3) unirse automáticamente por el componente de autoría 640 basándose en heurística algorítmica, y/o 4) incitar, por el componente de autoría

del autor y/o usuario, a especificar una unión cuando se determina que una unión no puede hacerse algorítmicamente.

La presente invención será modalizada en otras formas específicas sin apartarse de su espíritu o características esenciales.

5 Las modalidades descritas se van a considerar en todos los aspectos solamente como ilustrativas y no restrictivas. El alcance de la invención, por lo tanto, se indica por las reivindicaciones anexas en lugar de por la descripción anterior. Todos los cambios que vienen dentro del significado y escala de equivalencia de las
10 reivindicaciones serán abarcados dentro de su alcance.

15

20

25

REIVINDICACIONES

1. Un método para compilar una cadena de transformación de una interfase de usuario de recálculo, el método comprende:

5 un acto de analizar la cadena de transformación de la interfase de usuario de recálculo para dependencias para crear una gráfica de dependencia de dependencias entre entidades;

un acto de utilizar la gráfica de dependencia para crear un nivel inferior de pasos de ejecución; y

10 un acto de hacer la gráfica de dependencia disponible a un tiempo de ejecución para el programa.

2. El método de acuerdo con la reivindicación 1, en donde la gráfica de dependencia incluye una identificación de dependencia inter-entidad a partir de la cual se puede determinar que si una
15 primera entidad es evaluada, entonces una segunda entidad también va a ser evaluada.

3. El método de acuerdo con la reivindicación 1, en donde la gráfica de dependencia incluye una dependencia de evento a partir de la cual se puede determinar que si un evento de usuario ocurre,
20 entonces una entidad va a ser evaluada.

4. El método de acuerdo con la reivindicación 1, en donde la interfase de usuario de recálculo es un documento de hoja de cálculo.

5. El método de acuerdo con la reivindicación 1, en donde la
25 interfase de usuario de recálculo tiene un control complejo que tiene

parámetros de entrada y parámetros de salida de la cadena de transformación.

6. El método de acuerdo con la reivindicación 1, en donde la cadena de transformación se expresa declarativamente.

5 7. El método de acuerdo con la reivindicación 6, en donde el nivel inferior de los pasos de ejecución se expresa en un código de lenguaje imperativo.

8. Un producto de programa de computadora que comprende uno o más medios de almacenamiento legibles por computadora que
10 tienen instrucciones ejecutables por computadora que están estructuradas de manera que, cuando se ejecutan por uno o más procesadores de un sistema de cómputo, las instrucciones ejecutables por computadora hacen que el sistema de cómputo realice un método para compilar, que comprende lo siguiente:

15 un acto de analizar una transformación de una fuente de datos a un colector de datos en una interfase de usuario de recálculo;

un acto de crear una gráfica de dependencia de la transformación, la gráfica de dependencia representando una dependencia entre un evento y una entidad; y

20 un acto de utilizar la gráfica de dependencia para crear un nivel inferior de pasos de ejecución.

9. El producto de programa de computadora de acuerdo con la reivindicación 8, en donde el método para compilar además comprende:

25 un acto de hacer una gráfica de dependencia disponible para

un tiempo de ejecución para el programa.

10. El producto de programa de computadora de acuerdo con la reivindicación 8, en donde el colector de datos es un control.

5

10

15

20

25