



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 356 730**

51 Int. Cl.:
G06Q 10/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **02759443 .1**

96 Fecha de presentación : **26.08.2002**

97 Número de publicación de la solicitud: **1430409**

97 Fecha de publicación de la solicitud: **23.06.2004**

54 Título: **Método y sistema para producir una complicación ordenada de información con múltiples autores que contribuyen a la información en forma simultánea.**

30 Prioridad: **31.08.2001 US 944131**

45 Fecha de publicación de la mención BOPI:
12.04.2011

45 Fecha de la publicación del folleto de la patente:
12.04.2011

73 Titular/es: **Robert Tischer**
6905 Valley Brook Drive
Falls Church, Virginia 22042-4024, US

72 Inventor/es: **Tischer, Robert**

74 Agente: **Carvajal y Urquijo, Isabel**

ES 2 356 730 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Método y sistema para producir una compilación ordenada de información con múltiples autores que contribuyen a la información en forma simultánea

Antecedentes de la Invención

5 Con la llegada del ordenador personal, la capacidad de las personas de crear documentos se ha convertido en una tarea menos pesada. Hay un número de aplicaciones de edición de caracteres que existen en el mercado que permiten a las personas crear documentos, cada uno de estos productos variando sus capacidades. Estos productos ofrecen primariamente un contexto basado en soluciones. Donde estos productos padecen o carecen de una solución está la capacidad de generar un documento donde múltiples autores son necesarios o deseados para
10 compilar el documento. Más específicamente, estos productos no permiten múltiples autores para contribuir a la compilación de un documento en forma simultánea.

En muchas instancias, la capacidad de construir un documento electrónico puede ser a la vez una tarea que consume en cuanto a si el proceso es cumplido por un solo autor o por múltiples autores. Esta última proposición es exacerbada por los límites de la tecnología actual. Varias de las aplicaciones que existen generan un archivo de
15 datos electrónico que forma la base contextual para el contenido del autor. Este archivo usualmente está residente ya sea en un disco duro de un ordenador personal o en un servidor, el cual en esta instancia se pone a disposición a un conjunto de personas. Por ejemplo, el ambiente típico de un ordenador de negocios involucra una red de comunicación, la cual incluye estaciones de ordenadores de trabajo y servidores que actúan para integrar el sistema completo. Cuando un documento es ensamblado por primera vez, es almacenado en los servidores donde cualquier
20 usuario que tenga privilegios de acceso puede abrir el documento y proceder con el proceso de edición.

Los esfuerzos de dos o más usuarios para trabajar en forma simultánea son reunidos con frustración cuando se intenta en sistemas conocidos. Con la actual tecnología, una de las dificultades es que cuando un primer usuario está trabajando activamente en un archivo, el segundo usuario tiene que trabajar con dificultad en el mismo archivo al mismo tiempo. El segundo autor intenta añadir su sección al documento en el mismo momento así como otro
25 autor encontrará un número de escenarios diferentes. El primer escenario es que será denegado el acceso debido a que otro usuario ha abierto el archivo y el sistema solo permitirá tener acceso a un usuario a la vez. Otro atributo negativo de este sistema particular es que siempre y cuando el primer autor tenga el archivo abierto, si está trabajando en él o no, nadie más puede tener acceso al archivo.

Otro escenario es que el segundo autor es informado por el sistema que otro usuario tiene el archivo abierto. Esto resulta en la negación al archivo o el sistema sugerirá al segundo autor una pregunta su desea una copia "fantasma" del archivo o no. Si el segundo autor decide aceptar la copia "fantasma" y proceder a hacer los cambios, la aplicación no permitirá que la copia "fantasma" sea salvada en lugar del archivo original. Otras técnicas para la creación de un solo documento son más primitivas en naturaleza. Un grupo de autores pueden decidir en crear su propia sección en un archivo separado, el cual eventualmente puede ser fusionado en un solo documento. Con la
30 actual tecnología, el proceso de creación toma más tiempo y es inherentemente más complicado tomando en cuenta problemas de formato que usualmente atormentan la fusión de varios componentes incluso de formatos existentes pero más aun con diferentes formatos.

1.1 Técnica Relacionada

1.1.1 Patentes

40 1.1.1.1 Patente de los Estados Unidos 5,515,491

La patente de los Estados Unidos 5,515,491 divulga una colaboración de un sistema de procesamiento de datos. La colaboración del sistema de procesamiento de datos divulgado en la Patente de los Estados Unidos 5,515,491 opera en un ambiente de servidor cliente e intenta lograr su meta permitiendo a múltiples usuarios compartir un objeto en común. Una debilidad particular del sistema divulgada en la Patente de los Estados Unidos 5,515,491 es que el
45 sistema carece de una emulación manejable del ordenador del contexto del lenguaje y por lo tanto no puede distinguir entre un contexto del lenguaje y un contenido. Como tal, el sistema divulgado es reducido para tratar con caracteres y movimientos del cursor en los cuales el cambio en el elemento atómico más pequeño es el carácter. Por ejemplo, cuando el cursor de un usuario del sistema divulgado se posa en el comienzo de la palabra, toda la palabra se convierte en un área de bloqueo flotante. Ya que una palabra es una construcción del lenguaje natural arbitrario y no contribuye directamente a la estructura del documento, la implementación pierde de vista la meta de la creación en colaboración, la cual simula el ambiente de un solo autor.

50 1.1.1.2 Patente de los Estados Unidos 6,047,288

La Patente de los Estados Unidos 6,047,288 divulga una colaboración de un sistema de procesamiento de datos. La colaboración de un sistema de procesamiento de datos divulgado en la Patente de los Estados Unidos 6,047,288 opera en un ambiente de servidor cliente e intenta lograr su meta permitiendo a múltiples usuarios compartir un objeto en común. El sistema divulgado en la Patente de los Estados Unidos 6,047,288 no tiene un sentido de controlar los atributos del ambiente que permiten a los usuarios verificar en porciones de caja un objeto compartido. Sin embargo, la estructura de parámetros de control es una constante a lo largo de los documentos y consecuentemente no expresa el contexto lingüístico del documento. Por ejemplo, el sistema divulgado en la Patente de los Estados Unidos 6,047,288 sugiere un nodo de "sesión" de raíz con sub nodos como "notificar en inicio", "difundir", "título", etc. Dado, por ejemplo, un segundo documento, el nodo raíz y los identificadores de sub nodo podrían ser los mismos ("sesión", "notificar en inicio",...) excepto que sus valores podrían ser diferentes. El valor de "sesión" puede, por ejemplo, ser 3 en vez de 2. De ninguna manera pueden estos sub nodos ser construidos como una gramática en la forma en que la gramática es usada en la presente invención.

Sin una gramática lingüística coherente impuesta como el contexto a lo largo del conjunto de autores contribuyentes, la colaboración natural del sistema divulgada en 6,047,288 es reducida l contenido de arbitrariedad de un documento orientado al manejo de configuración del sistema cuyas piezas son simplemente verificadas una y otra vez. Una colaboración de una parte del objeto no es sinónimo con un documento virtual de múltiples autores.

1.1.2 Aplicaciones de Colaboración de Autores

1.1.2.1 Documentación

La documentación es un contenido de manejo del sistema. La meta del diseño guía de los sistemas de manejo de contenido es un modelo de cualquiera-a-cualquiera más el flujo de trabajo. Esto de hecho es un tipo de colaboración de documento, pero la colaboración no va más allá que el archivo. La documentación tiene numerosas instalaciones de búsqueda para manejo de contenido en archivos de cualquier formato, pero el modelo de servidor cliente y los sistemas basados en el archivo sufren de debilidad.

1.1.2.2 Notas Lotus

Las Notas Lotus sincronizan los archivos cliente en una base periódica a través de un acuerdo sobre el servidor. Las Notas Lotus es un servidor cliente y un archivo base y consecuentemente no tiene ningún concepto de distribuir el contexto lingüístico como la presente invención lo hace. Aunque las Notas Lotus colaboran, estas colaboran en diferentes entidades que el la presente invención lingüísticamente representa, el nodo.

1.1.2.3 Editores Estructurados

1.1.2.3.1 Generador Sintetizador

El generador sintetizador es una herramienta de edición estructurada lanzada por primera vez en 1981. El Generador Sintetizador trata con los lenguajes del ordenador y no con un contexto semántico lingüísticamente libre de gramática (véase 4.1 "contexto semántico libre de gramática"). El generador sintetizador no tiene capacidad de colaboración de red.

1.1.2.3.2 Grifón y Alianza

Grifón (Decouchant et al. 1993) y Alianza (Decouchant et al., 1995 y Decouchant et al. 1996) cada uno proporciona un ambiente de documento distribuido implementando controles de edición en la forma de varias partes de un documento a través del uso de SGML. Fragmentos de documentos son adquiridos, y allí hay un fragmento maestro por sistema. Cuando un fragmento maestro es revisado, cambiado y nuevamente verificado, fragmentos esclavos son notificados y actualizados.

Sin embargo, cada uno de estos sistemas está limitado en que cada uno del conjunto de fragmentos es estático en todo el ciclo de edición, el cual es contrario a como son escritos los documentos en la práctica. Los elementos del SGML no están estrictamente atados ya sea a los fragmentos de noción Grifón o Alianza. Consecuentemente, Grifón y Alianza son ineficientes en su capacidad de mantener toda la consistencia del documento. Además, ni Grifón ni Alianza soportan un cambio en la estructura de un documento después del inicio (excepto tal vez exportándolo y reiniciándolo) debido a que estos programas dependen en estructuras que puedan solamente ser construidas al inicio. Ni Grifón ni Alianza tienen instalaciones punto a punto para cambio de contenido o para el papel y privilegio de almacenamiento y aplicación. Grifón y Alianza son generalmente editores estructurados con mecanismos persistentes, los cuales siempre fuerzan la centralización al menos en una parte, del documento distribuido. Los papeles y los privilegios no están atados a la estructura del documento.

En la patente de los Estados Unidos 6,088,702, un sistema de grupos de publicación es descrito con un almacenamiento de documentos centralizado. Si un usuario accede a un documento, el documento es bloqueado para otros usuarios. El sistema registra todos los cambios y un editor maestro es invocado para una revisión administrativa final. Cada usuario del sistema tiene la capacidad de comunicación electrónica a través del sistema con cualquier otro usuario del sistema. En este sistema, todos los cambios son fusionados por un operador maestro.

También en "CCDES – un sistema de edición de documento compuesto colaborador", por Chien-Hung Lin, Computer Communications 19 (1996), páginas 590 – 596, y en la WO/0062220 sistemas de edición de grupo son descritos con una fusión de documento y almacenamiento de documento centralizado. En estos documentos, usuarios individuales contribuyen a un documento, el cual es compilado por una institución central, por ejemplo un operador. Los sistemas divulgados en estos documentos pasivamente asisten durante el proceso de compilación proporcionando y representando la información de documento almacenada de una manera apropiada.

En la patente de los Estados Unidos 5,245,553, es descrito un sistema de comunicación, en el cual un usuario terminal puede ver la misma porción de un documento en común. Si se hacen cambios al mismo tiempo por usuarios distintos, los usuarios pueden interactuar con la pantalla. En particular, si los usuarios están viendo la misma porción, cada usuario puede interactuar con otros usuarios a través de la pantalla sin tener que esperar a un usuario a que complete su participación. Esto proporciona una comunicación interactiva entre los usuarios trabajando en la misma porción de documentos. Consecuentemente, el sistema de comunicación permite cambios simultáneos dentro de la misma porción de documento y son resueltos conflictos de entrada de diferentes usuarios concernientes a la misma porción de documento por una comunicación directa entre los usuarios. Esto, sin embargo, no prohíbe los conflictos de entradas dentro de la misma porción de documento, por ejemplo si un usuario diferente no está de acuerdo en una sola entrada en común. Además, la patente de los Estados Unidos 5,245,553 no divulga el uso de DTD.

La patente de los Estados Unidos 6,061,697 divulga un aparato de manejo de documento en el cual el DTD es un trabajo de edición que es ejecutado en paralelo. Son provistas restricciones para edición parcial del DTD. Una definición de tipo de edición parcial de documento es formada por la modificación de la definición tipo del documento original de todo un documento. Una unidad de acceso a un documento SGML ingresa y saca una parte o de todo un documento SGML almacenado en una base de datos correspondiente a una solicitud del cliente. Además, la unidad de acceso al documento SGML, crea una edición parcial del DTD, maneja las revisiones del documento SGML, y guarda/restaura el documento SGML. Así, la patente de los Estados Unidos 6,061,697 describe como proporcionar copias para usuarios en particular en las cuales los usuarios trabajan e integran las partes editadas dentro del documento principal después de que la edición es completada. Por lo tanto, es descrita una estructura centralizada, en la cual un editor sólo trabaja en las copias y no trabaja en el mismo documento en evolución original. Además, la estructura centralizada bloquea partes completas (las cuales son enviadas al editor solicitado) y la restricción de información es mantenida y aplicada por un ente central, el cual también mantiene el documento en evolución original.

Como se puede ver, un documento con múltiples autores es difícil de ensamblar de acuerdo a los sistemas y métodos conocidos. La actual tecnología no permite una forma eficiente, y en algunos aspectos, crea una angustia adicional con su limitación y sus capacidades. Así, existe la necesidad de una aplicación que permita más de un autor que contribuya a un documento: en forma simultánea mientras permite varios autores, así como observadores del proceso, ver como se completan los cambios o las actualizaciones.

2 Resumen de la invención

De acuerdo con esta y otras necesidades, se provee de acuerdo con la presente invención un método y un sistema de acuerdo a la reivindicación 1 y 12 respectivamente para producir un documento distribuido teniendo una compilación de información ordenada. El sistema comprende un editor de fragmentos múltiples ejecutables que funcionan, cooperativamente como una declaración del tipo de documento implementada (DTD). Cada uno de los ejecutables del editor de múltiples fragmentos es un nodo específico de la implementación del DTD. Los ejecutables del editor de múltiples fragmentos están distribuidos entre múltiples sitios de la red de ordenadores y operan en un ambiente punto-a-punto sin la necesidad de un servidor central. El sistema permite múltiples autores para editar un documento distribuido en forma simultánea mientras permite a cada uno de los múltiples autores ver las ediciones hechas por otros de los múltiples autores.

En una realización del sistema, las repeticiones del documento completo residen en múltiples sitios de la red de ordenadores.

En una realización del sistema, una edición hecha por cualquier otro de los múltiples autores es propagada entre las repeticiones que residen en los múltiples sitios de la red de ordenadores.

En una realización del sistema, una edición hecha por cualquier otro de los múltiples autores es inmediatamente propagada entre las repeticiones que residen en los múltiples sitios de la red de ordenadores.

En una realización, una edición hecha por cualquier otro de los múltiples autores es retrasada antes de ser propagada entre las repeticiones que residen en los múltiples sitios de la red de ordenadores.

- 5 En una realización, una edición hecha por cualquier otro de los múltiples autores es retransmitida para intervención de personal, y así, es retrasada, antes de ser propagada entre las repeticiones que residen en los múltiples sitios de la red de ordenadores.

En una realización, una edición hecha por cualquier otro de los múltiples autores es propagada como una transacción atómica entre las repeticiones que residen en los múltiples sitios de la red de ordenadores.

- 10 En una realización, las repeticiones residen en una memoria de ordenador en los múltiples sitios de la red de ordenadores.

En una realización, las repeticiones se conservan escribiendo en discos duros de ordenador en los múltiples sitios de la red de ordenadores.

- 15 En una realización, cada autor de los múltiples autores es asignado a un tipo de entidad que tenga asociada con él correspondiente a los ejecutables que profanen el papel que un autor pueda tener en la creación de un documento distribuido.

En una realización, los papeles que un autor pueda tener en la creación de un documento distribuido incluyen: contexto raíz del autor, contexto del autor, y contenido del autor.

En una realización, un suscriptor puede ver ediciones hechas por uno o más de los múltiples autores.

- 20 La presente invención opera en un ambiente punto-a-punto. Además, la presente invención genera ejecutables basados en gramática para cualquier DTD (véase 4.2 “declaración de documento tipo”), los cuales son la base para la edición cooperativa. Por otra parte, la presente invención utiliza una emulación manejable de ordenador de un contexto del lenguaje para hacer explícita una estructura de un documento común, y así poder distinguir entre el contexto del lenguaje y el contenido. Además, ya que la presente invención es el elemento más pequeño de cambio
25 es una entidad libre del contexto de gramática semántica indivisible, la presente invención sabe que múltiples cambios en un elemento de gramática del documento son irrelevantes antes que sean entregados.

- La presente invención enlaza toda actividad a la estructura gramática, la cual se permite a sí misma evolucionar en todo el ciclo de vida del documento. Por supuesto, el contenido creado utilizando la presente invención cambia con respecto a los nodos de contexto de gramática actuales. Las señales estructurales como son implementadas en la
30 presente invención no tienen que ser simbólicas y pueden ser un texto claro y son así indistinguibles del contenido de texto (esto es una cosa buena). Los papeles y los privilegios como son implementados en la presente invención son enlazados a la gramática y ocurren como una creación de nodos descendiente adicional (los nodos pueden ser eliminados o también modificados). Esto hace que cada parte de un documento creado utilizando la presente invención, por definición, coherente con su estructura gramática inherente.

- 35 La presente invención no tiene necesidad de un servidor centralizado o no tiene que haber la persistencia de un disco duro ya que rellenar un documento ocurre de los familiares del nodo en vez de un almacenamiento central. El modo por defecto de la presente invención es una respuesta de la totalidad del documento en cada sitio. Para la presente invención, solo señales visuales diferencian las porciones estructurales del documento, las cuales prestan a sí mismo un comportamiento de escritor natural. Un documento distribuido creado utilizando la presente invención
40 es realmente un conjunto completo de implementaciones de nodo de gramática cada uno implementado para el propósito expreso de edición que el trozo en particular del documento en lugar de ser varias copias de editores estructurales generalizados.

2.1. Creación de Redes

- 45 Una ventaja de la invención es la capacidad de enviar cambios logrados por un autor en el momento de su ocurrencia. Los cambios son puestos en la red de todos los otros participantes del documento cuando un autor señala que el cambio está completo. La finalización de cambio de contenido ocurre cuando el autor utiliza implícitamente una de las señales de estilización de la interfaz de usuario o las acciones definidas cuando la interfaz de usuario fue definida (véase 5.4.2.2 Interfaz de Diseño de Usuario).

En una realización, los cambios de contexto y contenido son siempre puestos inmediatamente a todos los otros participantes del documento haciendo que la banda ancha utilizada sea directamente proporcional a la cantidad de cambios teniendo lugar en el documento distribuido en ese momento. En otra realización, los cambios de contexto y contenido son retransmitidos a través del personal involucrado también llamado flujo de trabajo, antes de llegar a su destino.

Los cambios de contexto y contenido propagados no deben interrumpir los beneficiarios cuando sea recibido y están al mismo tiempo añadidos a los procesos de ejecución de los beneficiarios. La interrupción más severa a un beneficiario es reemplazada con el ejecutable del beneficiario ocasionando una acción de repoblación. La invención prescribe que, dadas las alternativas de implementación, la alternativa que menos afecta a los usuarios debe ser elegida con el fin de preservar la apariencia de edición de un solo documento.

Ya que el control de cambios es jerárquicamente distribuido y la autoridad de cambio de contenido es mutuamente exclusiva y exhaustiva, la familia o las condiciones de punto muerto no pueden ocurrir a lo largo del caso del documento de la red.

2.2 Respuesta del Documento

La invención proporciona una réplica equivalente del documento en el sitio del nodo de cada autor y en cada sitio de un suscriptor. Cada nodo contiene un documento similarmente construido y por lo tanto cada nodo mantiene la versión actual completa del documento por defecto. Ya que por cambios en el diseño del documento por los autores no ocasionan problemas de concurrencia, un cambio en la transacción no necesita ser causalmente secuencial, y por lo tanto puede llegar y ser aplicada en cualquier orden en sus destinos.

Con el fin de mantener la réplica invariable, cada transacción debe, sin embargo, ser atómica, asegurando así que las instancias del documento en estado inactivo en cada sitio son réplicas exactas. Para un mensaje de cambio a ser atómico, el ejecutable del beneficiario debe eventualmente reconocer que este ha recibido y aplicado el cambio solicitado. Aunque no hay límite de tiempo en cuanto pueda ser aplicada la solicitud de un cambio, la solicitud simplemente no es un mensaje de difusión sin conexión que no necesita reconocimiento. Un documento está sólo inactivo si el contexto raíz del autor detiene a todos los autores de realizar cambios y todos los mensajes de cambios han sido entregados satisfactoriamente.

2.3 Poblar

En el sitio ejecutable del ordenador está codificado para poblar por su propia instancia al inicio o ya sea en el momento de creación o después del apagado. Ya que todos los sitios por defecto contienen réplicas exactas, un sitio donante es elegido aleatoriamente para poblar la instancia recientemente creada. La invención permite para cualquier algoritmo de población ser usado siempre y cuando el algoritmo no sea 100% dependiente del almacenamiento en el disco, el cual por la definición está siempre inconsistente con el documento distribuido virtualmente.

2.4 Documento Generalizado

El uso en la presente invención del término documento es más amplio que un simple texto secuencialmente organizado.

Los documento son escritos por lo que una mente humana puede empezar percibiendo y entendiendo al comienzo del documento y continuar este proceso hasta que el final del documento sea alcanzado. Esto es un proceso temporal inherentemente semántico.

La presentación, sin embargo, de la instancia de cualquier nodo de autor o suscriptor pueda tomar cualquier forma siempre y cuando mantenga invariable su semántica temporal. Por ejemplo, en vez de párrafos, pueden ser llevadas a cabo secciones y palabras como son descritas en la ilustración de la Fig. 7, clips de animación, el resultado de los cuales puede ser un formato de presentación de imagen en movimiento. La gramática semántica subyacente (véase 4.1 "contexto libre de gramática semántica") puede quedar igual.

2.5 Imitación del Lenguaje Natural

La meta de la presente invención es crear un ambiente de ordenador que facilite que un documento sea autorizado cooperativamente y simultáneamente por múltiples autores sin perder la integridad del documento. Este proceso recapitula la formación y la evolución del lenguaje natural. Este proceso de formación y evolución es uno del contexto complementario y de la creación y cambio de contenido. La presente invención emula este proceso de creación de lenguaje natural organizando un software de ordenador y estructurando la cooperación del documento a

lo largo de estas mismas líneas. Por lo tanto, el contexto y la creación de contenido están siempre en actividades separadas.

5 Ya que la creación y el cambio del lenguaje natural es el documento múltiplemente editado en el mundo real, entonces la representación manejable de ordenador de este proceso, el cual es la presente invención, es por lo tanto el documento múltiplemente editado utilizando software de ordenador.

2.6 Nodo de Base, No Archivo de Base, Invalidación Falsa Minimizada

10 El archivo en un directorio de ordenador tiene un solo nombre y contenido. Si alguien cambia cualquier carácter en ese archivo, hace que cualquier copia del archivo sea inválida. A corto de mantener los punteros a piezas individuales o caracteres en el archivo, no hay manera de decir cual pieza del documento ha invalidado todo el archivo. Los sistemas del documento y los procesadores son todos basados en el archivo o controlados en concurrencia con el fin de mantener la integridad del documento, la cual impone una interferencia del lenguaje no natural en el proceso de creación del documento, Haciendo un cambio en cualquier lugar en un archivo falso invalida las áreas que no han sido cambiadas.

15 La presente invención enlaza la integridad del documento con el nodo semántico atómico en vez del archivo. Así, cuando un cambio es hecho en cualquier lugar en el contenido de un nodo semántico, solo ese nodo atómico se vuelve inválido con respecto a la réplica de nodos en otros lugares. El resultado general es un conjunto de mutuamente excluyentes aún áreas de trabajo de documento semántica atómica, cuya suma semántica es todo un tema coherente. Cuando los nodos atómicos para un contexto semántico han sido predeterminados como es el caso en la presente invención, no hay invalidación falsa.

20 2.7 Estilo de Escritura Base Promedio; No Etiquetas; No WYSIWYG

25 El marcado de lenguajes permite a la estructura del documento implícita convertirse en un ordenador procesable. Pero la estructura y el contexto se suponen que son invisibles cuando se produce un contenido de lenguaje natural. Así, la etiqueta de artefacto por su presencia es un obstáculo. Las aplicaciones de documento actuales usualmente muestran estas etiquetas como árboles que pueden ser visibles cuando el autor necesita orientación. O algunas veces, el actual conjunto de etiquetas legales aparecen como listas desplegables desde las cuales el autor puede elegir. El problema es que estos artefactos de control de ordenador estorban con el proceso de edición.

30 La presente invención pide para artefactos de documento estándar a ser usados como un estilo representativo o contexto inmutable, como diferenciadores de contexto y contenido, y como indicadores de movimiento desde un nodo semántico a otro por un autor. Un ejemplo de las corrientes de requerimientos que estorban en el proceso de edición es la tecnología WYSIWYG (lo que ve es lo que obtiene). Aquí los requerimientos de procesos de impresión de un solo usuario se imponen directamente en el proceso de creación del documento.

2.8 Persistencia Distribuida

35 Una consecuencia de esta invención es, una réplica de documentos existe en cada edición o sitio de suscriptor, y no hay necesidad de que el documento sea escrito en un disco duro, esto es, persistencia. Los sitios de documento son poblados en el inicio desde un conjunto aleatorio de sitios hermanos y el resultado promedia el tiempo entre las fallas sólo por encima de que unos pocos sitios se conviertan en mayor que los discos duros. El disco duro se vuelve irrelevante. Por supuesto, no hay problema de persistir en el documento desde que cualquier sitio tenga la capacidad de exportar su aplicación que está corriendo como una instancia marcada.

2.9 Punto-a-Punto; Sin Cliente Servidor

40 Los documentos réplica en cada sitio significan que no hay servidor central que pueda crear un cuello de botella. El flujo de información es completamente dictado por las necesidades y requerimientos semánticos y de cooperación. La presente invención es verdaderamente punto-a-punto debido a que no hay la necesidad de imposiciones de concurrencia de datos.

2.10 Marcado para edición, no Recolección de Datos y Dispersión

45 El marcado es usado hoy en día principalmente para agregar y sindicarse la información que ha sido marcada con marcas que parecen lenguaje natural, como <vegetales>. La mayor deficiencia en esta aproximación es que se asume que los datos ya han sido creados ya sea por un solo autor o por una serie de autores individuales. En contraste, la presente invención disemina durante el proceso de creación, el cual es un super conjunto del modelo de agregación-sindicación y cubre el ciclo de vida real y completa del documento. La razón por la que el actual
50 proceso de marcado no incluye el proceso de edición es, que no hay funcionalidad en el modelo de agregación

5 sindicación para la cooperación de los DTDs (véase 4.2 “declaración de documento tipo”), el cual representa los contextos lingüísticos del que es agregado y sindicado. Por lo tanto, una instancia de cooperación sobre la cual es aplicada la agregación y la sindicación, es una cacofonía verdadera de los procesos de búsqueda de ordenador. Estas instancias DTD dispares y por lo tanto relegadas a tener que ser buscadas y extraídas, las cuales tienen muy poco o nada que hacer con el actual proceso de edición.

2.11 Editor Estructurado No Generalizado, No Servidor Cuello de Botella

10 La presente invención no es un editor generalizado más bien un juego de editor de fragmentos ejecutables complementarios generados por documento. La presente invención no es un editor generalizado pero en lugar es un juego de fragmentos de editores ejecutables complementarios generados por documento. Múltiples fragmentos de editor ejecutables son creados para que funcionen cooperativamente como un DTD implementado (véase 4.1 “declaración de documento tipo” y 4.30 “fragmento de editor ejecutable”). Cada fragmento de editor ejecutable es en efecto un nodo específico de implementación racionalizado del DTD. La diferencia en la eficiencia utilizando cualquier tipo de medición es alta. También, desde que la presente invención opera punto-a-punto con una réplica del documento completo en cada sitio, el ambiente de edición resultante para la presente invención automáticamente alcanza la meta de vista y sentimiento de un solo autor, que no es posible con el modelo de servidor cliente el cual solo presenta al autor el nodo o nodos verificados para la tarea de edición a la mano. Los demás nodos verificados en la aproximación de servidor cliente, los otros autores tendrán que esperar a hacer sus cambios. Incluso si la actual tecnología de documento de servidor cliente se amplía para permitir sólo la lectura de verificación para facilitar a un solo autor ver y sentir, no habría inherentes formas para las porciones de sólo lectura a ser actualizadas. Aunque es posible utilizar la devolución de llamado para actualizar los clientes en el modelo de servidor cliente, es extremadamente intensiva la aproximación de programación y simplemente podría servir además para dividir las actividades involucradas en la edición colaborativa sin costura.

25 La presente invención asegura que hay una constante evolución exhaustiva y mutuamente un conjunto exclusivo de autores que cooperativamente se apropian y desarrollan el documento virtual. Esto es solo teóricamente posible utilizando los privilegios de base de datos tradicional. Los privilegios, la propiedad y los comportamientos de ejecución de los nodos en el modelo de documento de servidor cliente son actividades separadas, donde estas características en la presente invención son indivisibles como sea necesario para la integridad virtual del documento.

El modelo punto-a-punto de la presente invención ha llevado a una colaboración colaborativa eficiente de la aplicación de edición del documento sin la necesidad de un servidor de cuello de botella.

30 Características y ventajas de objetos adicionales de la invención serán aparentes en la descripción detallada a continuación.

3 BREVE DESCRIPCIÓN DE LOS DIBUJOS

35 Los aspectos anteriores y muchas de las ventajas acompañantes de esta invención serán más fácilmente apreciadas así como se vuelven de mejor entendimiento con referencia a la siguiente descripción detallada, cuando se toman en conjunción con los dibujos acompañantes, donde:

La FIG. 1 es una representación gráfica de la generación del contexto raíz de los autores.

La FIG. 2 es una vista jerárquica de la presente invención con respecto al contexto del autor creando dos sub nodos.

La FIG. 3 es una vista esquemática de la FIG. 2 en estado de ejecución.

40 La FIG. 4 es una vista jerárquica de la presente invención con respecto a un contexto subordinado del autor además de la creación de dos sub nodos.

La FIG. 5 es una vista esquemática de la FIG. 4 en estado de ejecución.

La FIG. 6 es una vista de un diálogo de marco representativo para la asignación de nodos de la invención;

La FIG. 7 es una vista en pantalla de un nodo ejecutable de la invención en estado de ejecución.

La FIG. 8 es una vista de pantalla de la FIG. 7 exportada como una instancia SGML regular sin presentación.

45 4 DEFINICIONES

4.1 “contexto gramático libre de semántica”

5 Como es usado aquí y en las reivindicaciones, la frase “contexto gramático libre de semántica” es un término en lingüística, “gramática semántica”, combinado con un término en la ciencia de ordenador, “gramática libre de contexto”, o CFG. Los dos términos componentes son análogos. Los CFGs consisten de no terminales definidos por producciones, los cuales son además definidos por terminales. Por lo tanto el siguiente es una gramática completa de la perspectiva de la ciencia de ordenador (las letras mayúsculas representan no terminales y las letras minúsculas representan las terminales).

A:BcDe

B:b

10 D:d

15 Las gramáticas semánticas en las lingüísticas están evolucionando en estructuras del lenguaje natural, algunas veces llamadas paradigmas, teniendo con vida mediante la práctica de los miembros definiendo grupos de humanos lingüísticamente competentes. Ejemplo de gramáticas semánticas explícitamente estandarizadas son los conjuntos de transacciones y mensajes de EDI (intercambio de datos electrónico). Otro ejemplo es el Manual Merck (véase 5.6 “Ilustración de Evolución de Documento”). Esta gramática semántica particular es definida presumiblemente por un conjunto de doctores médicos autorizados, y cuya estructura puede ser implementada en un ordenador utilizando técnicas de análisis de ciencia de ordenador estándar.

20 La presente invención equivale a un nodo de gramática semántica con CFGs de no terminal e implementa gramáticas semánticas en ordenadores de una forma que permite la semántica de no terminales para comportarse como el contexto de los autores de la presente invención (véase 4.17 “contexto del autor”) y terminales para comportarse como los contenidos de los autores de la presente invención (véase 4.18 “contenido del autor”) con el fin de colaborar en la creación de contenido lingüístico socavando el proceso de expresión creativo.

4.2 “declaración de documento tipo”

25 Como es usado aquí y en las reivindicaciones, la frase “declaración de documento tipo” (DTD) se refiere a las reglas estándar ISO 8879 para la producción de sintaxis del documento. Estas sintaxis del documento tienen el potencial de ser humanamente leíbles y por lo tanto capaces de dar el sentido del lenguaje natural al creador del documento. El ISO 8879 asegura que la sintaxis del documento producido son ordenadores tratables garantizando así un vínculo entre el lenguaje natural y su representación como el contexto en un ordenador.

4.3 “compilación ordenada de información”

30 Como es usado aquí y en las reivindicaciones, la frase “compilación ordenada de información” se refiere al texto escrito de las animaciones secuenciadas de tal manera que imparten conocimiento coherente a un humano alfabetizado percibiendo temporalmente intentando entender la secuencia desde el principio hasta el fin.

4.4 “contexto”

35 Como es usado aquí y en las reivindicaciones, el término “contexto” se refiere al ambiente anterior dentro del cual es producido el contenido escrito o hablado de un lenguaje natural. Aunque hay muchas partes del contexto del ambiente del lenguaje natural, el ISO 8879 resalta la jerarquía semántica que es más relevantemente representativa. El DTD hace el contexto del lenguaje natural explícito, y por lo tanto manejable al ordenador. El contexto lingüístico es siempre constante cuando el contenido es producido o modificado. Así, en el fragmento ejemplo:

<ManualMerck>El manual Merck apareció por primera vez en 1988 como un

40 texto fino de 262 páginas titulado Manual Merck para la Materia Médica.

<InsuficienciaCardiaca>

Un síndrome común que puede ser causado por varias etiologías

cuyas manifestaciones clínicas reflejan una anormalidad fundamental –

una disminución en el estado contráctil del miocardio tal como una

salida cardiaca (CO) es inadecuada para las necesidades del cuerpo.

</InsuficienciaCardiaca>

</ManualMerck>

5 Las piezas <ManualMerck>, </ManualMerck>, <InsuficienciaCardiaca>, e </InsuficienciaCardiaca> son la representación de conjunto de contexto lingüístico en alguna jerarquía semántica, donde la segunda pareja es semánticamente subordinada a la primera pareja.

4.5 “contenido”

Como es usado aquí y en las reivindicaciones, el término “contenido” es ejemplificado por el texto entre los pares de marcas en el “contexto” de definición anterior.

10 4.6 “ejemplo de etiquetas”

15 Como es usado aquí y en las reivindicaciones, la frase “ejemplo de etiqueta” es la expresión de ordenador manejable del contexto y es ejemplificada por las piezas <ManualMerck>, </ManualMerck>, <InsuficienciaCardiaca>, e </InsuficienciaCardiaca> en la definición del contexto anterior. Un “ejemplo de etiqueta” se refiere exclusivamente a una instancia SGML de un DTD (véase 4.2 “declaración de documento tipo”), específicamente el inicio y opcionalmente el fin del contenido envolvente de la etiqueta.

4.7 “transacción”

Como es usado aquí y en las reivindicaciones, el término “transacción” se refiere al ejercicio de un cambio de protocolo de mensaje de solicitud de un autor o un beneficiario. Este protocolo incluye hacer la solicitud, manejar los conocimientos y eventualmente cometer los resultados como irreversibles.

20 4.8 “transacción atómica”

25 Como es usado aquí y en las reivindicaciones, la frase “transacción atómica” se refiere a la transacción en la cual, si cualquiera de las partes del protocolo no son llevadas a cabo propiamente y llevando así a réplicas de documentos inconsistentes en varios sitios, entonces son tomadas acciones para reparar el beneficiario afectado. Recibiendo un conocimiento desde un beneficiario no necesariamente significa que el cambio ha sido aplicado, más bien que la aplicación del beneficiario promete a cualquier aplicar el cambio eventualmente o de lo contrario resolver el cambio. Esto permite para cualquier número de escenarios de flujo de trabajo que impliquen múltiples entregas de saltos de mensajes.

4.9 “sitio del suscriptor”

30 Como es usado aquí y en las reivindicaciones, la frase “sitio del suscriptor” se refiere al ordenador que está ejecutando una aplicación de un suscriptor en particular.

4.10 “sección de documento”

Como es usado aquí y en las reivindicaciones, la frase “sección de documento” se refiere a un nodo de representación visual en particular de un contexto y un contenido estilizados.

4.11 “recordación”

35 Como es usado aquí y en las reivindicaciones, la frase “recordación” se refiere a una lista de direcciones de sitios de tipo editor en particular a la que el contenido y las solicitudes de cambios administrativos nos enviadas cuando ellas ocurren.

4.12 “tipo de entidad”

40 Como es usado aquí y en las reivindicaciones, la frase “tipo de entidad” se refiere a las funciones particulares equipadas con sus correspondientes archivos ejecutables en la creación de un documento múltiplemente editado.

4.13 “tipo de editor”

Como es usado aquí y en las reivindicaciones, la frase “tipo de editor se refiere a los autores cuyos ejecutables les permiten al menos hacer cambios de contenido en el documento.

4.14 “tipo de suscriptor”

5 Como es usado aquí y en las reivindicaciones, la frase “tipo de suscriptor” se refiere a los participantes de un documento cuyos ejecutables les permiten al menos ver el documento.

4.15 “generador raíz de contexto”

Como es usado aquí y en las reivindicaciones, la frase “generador raíz de contexto” se refiere al software que puede crear y desplegar los autores del contexto raíz inicial. El generador de contexto raíz es el generador editor de fragmento de la presente invención (véase 4.32 “ejecutable del generador editor de fragmento”).

10 4.16 “autor del contexto raíz”

Como es usado aquí y en las reivindicaciones, la frase “autor del contexto raíz” se refiere a un ejecutable del tipo editor que represente el nodo superior en una jerarquía semántica de un documento.

4.17 “contexto de autor”

15 Como es usado aquí y en las reivindicaciones, la frase “contexto del autor” se refiere a un ejecutable del tipo editor que es semánticamente subordinado en el contexto raíz del autor y opcionalmente subordinado a otros contextos de autores, y quien tiene la capacidad de cambiar el contexto del documento así como el contenido.

4.18 “contenido del autor”

20 Como es usado aquí y en las reivindicaciones, la frase “contenido del autor” se refiere a un ejecutable de tipo editor que es semánticamente subordinado al contexto raíz del autor y opcionalmente a otros contextos de otros autores, quien no tiene nodos secundarios subordinados, y quien tiene la capacidad de cambiar el contenido del documento.

4.19 “suscriptor”

Como es usado aquí y en las reivindicaciones, la frase “suscriptor” se refiere a un ejecutable de tipo no editor que no tiene un nodo semántico asociado con él. Los tipos de suscriptor pueden ser creados por tipos de editor así como por otros suscriptores si es permitido.

25 4.20 “nodo”

30 Como es usado aquí y en las reivindicaciones, el término “nodo” se refiere a un elemento SGML que es un identificador simple del lenguaje natural representando una categoría de semántica lingüística, y también se refiere a un ejecutable que fue creado específicamente para llevar a cabo operaciones que son específicas para y consistentes con la categoría semántica que representa. Un nodo es por lo tanto una dualidad de la representación y la operación (véase 2.6 “Nodo Basado, No Archivo Base; Invalidación Falsa Minimizada”)

4.21 “nodo objetivo”

Como es usado aquí y en las reivindicaciones, la frase “nodo objetivo” se refiere a un nodo ejecutable creado por un autor de contexto existente o un autor de contexto raíz.

4.22 “sub nodo”

35 Como es usado aquí y en las reivindicaciones, la frase “sub nodo” se refiere a un nodo cuyo contenido semántico es subordinado al nodo actual.

4.23 “nodo descendente”

Como es usado aquí y en las reivindicaciones, la frase “nodo descendente” se refiere a y es intercambiable con un sub nodo.

40 4.24 “replicar vista de nodo”

- 5 Como es usado aquí y en las reivindicaciones, la frase “replicar vista de nodo” se refiere a un nodo que es exactamente el mismo que un nodo semántico con el mismo identificador en otro sitio en el mismo documento. Cada ejecutable de tipo editor de cada sitio es una implementación de un nodo de su propiedad, pero que también contiene réplicas de otros nodos que no son de su propiedad o pueden cambiar, los cuales por lo tanto son para vista.
- 4.25 “contexto de nodo del autor”
- Como es usado aquí y en las reivindicaciones, la frase “contexto de nodo del autor se refiere al ejecutable de su propiedad desplegado y usado por su autor de contexto.
- 4.26 “contenido de nodo del autor”
- 10 Como es usado aquí y en las reivindicaciones, la frase “contenido de nodo del autor se refiere al ejecutable de su propiedad desplegado y usado por su autor de contenido.
- 4.27 “comportamientos de escritura convencionales”
- 15 Como es usado aquí y en las reivindicaciones, la frase “comportamientos de escritura convencionales” incluye la capacidad de guión, esquema, resumen, ilustración, partición del texto en párrafos secuenciales coherentes, secciones y capítulos, así como las capacidades de elegir y utilizar señales visuales estructurales comunes que señalan la estructura del documento, tales como señales incluyendo pero no limitadas a señales de textos visuales tales como sangrías, numeración, sombreado, coloración, negrita, cursiva y subrayado.
- 4.28 “editor estructurado”
- 20 Como es usado aquí y en las reivindicaciones, la frase “editor estructurado” se refiere a un procesador de palabras que utiliza un DTD en vez de un carácter individual para guiar al usuario en el proceso de edición.
- 4.29 “editor estructurado generalizado”
- 25 Como es usado aquí y en las reivindicaciones, la frase “editor estructurado generalizado” se refiere a un editor estructurado que tiene la capacidad de aceptar, compilar y utiliza una gran variedad de estructuras, el más popular de los cuales es la capacidad de utilizar cualquier tipo de DTD (véase 4.2- “declaración de documento tipo”). La presente invención no emplea un editor estructurado comercialmente generalizado, más bien es un juego de editores de fragmentos complementarios cooperativos (véase 4.30 “fragmento de editor ejecutable”) por documento cada fragmento el cual es generado como sea necesario.
- 4.30 “fragmento de editor ejecutable”
- 30 Como es usado aquí y en las reivindicaciones, la frase “fragmento de editor ejecutable” se refiere a un ejecutable para operar en un nodo DTD en particular.
- 4.31 “fragmento de editor generador ejecutable”
- Como es usado aquí y en las reivindicaciones, la frase “fragmento de editor generador” se refiere a un código ejecutable que pueda crear ejecutables de fragmentos de editor (véase 4.30 “fragmento de editor ejecutable” para operación en nodos DTD en particular.
- 35 4.32 “repositorio”
- 40 Los repositorios son cualquier tipo de bases de datos que tiene una superestructura de lenguaje marcada. La razón que la industria eligió la palabra repositorio en vez de solo base de datos es, el propósito de la bases de datos es garantizar la integridad de los datos internos. Aunque los repositorios tienen la capacidad de indicar que los componentes y los elementos marcados han sido verificados en la base de datos, esto no garantiza su consistencia con respecto al resto de la información de base de datos interna. Las actualizaciones en los repositorios son llevadas a cabo en el cliente, donde las actualizaciones son llevadas a cabo en el servidor.. El nombre repositorio significa esta diferencia, lo que la hace más útil que un sistema de manejo de configuración de software.

DESCRIPCIÓN DETALLADA DE LAS REALIZACIONES PREFERIDAS

Una realización preferida de la presente invención es una aplicación de sintaxis directa que comprende un generador de contexto raíz y un autor de contexto raíz, y opcionalmente comprende uno o más seleccionados de un grupo consistente de un autor de contexto, un autor de contenido y un suscriptor. Con todos los componentes ejecutándose en conjunto, la aplicación funciona como un procesador de datos distribuido basado en el lenguaje. La invención permite a cada autor trabajar en su sección del documento mientras la vista concurrentemente cambia los otros autores trabajando en el mismo documento virtual.

5.1 El Papel del SGML

Toda expresión lingüística, donde es expresada por oradores, lectores o escritores, es producida dentro de un contexto lingüístico implícito. Con la llegada del Lenguaje Marcado Generalizado Estándar (SGML) ISO 8879 en 1986, la expresión lingüística en el ordenador se ha convertido en un proceso sin derechos de propiedad. Utilizando el SGML, la estructura semántica implícita de cualquier lenguaje, expresión o compilación natural puede ser expresada a través de la sintaxis de una Declaración de Documento Tipo (DTD). El DTD transforma el contexto lingüístico de la expresión o compilación de lenguaje en una forma explícita que es manejable en ordenador.

El SGML es un conjunto genérico de palabras clave, sintaxis de nomenclatura, ocurrencia y términos de conector estructurados de tal forma que el diseñador de la estructura del documento utiliza estos para expresar literalmente cualquier conjunto del mundo real de nociones semánticas del lenguaje natural en una notación que los ordenadores pueden procesar. El DTD es la expresión coherente procesable por un ordenador de tal lenguaje. El SGML puede producir un número infinito de DTDs.

En una realización preferida de la invención, múltiples autores cooperativamente abarcan un solo SGML DTD. Este DTD común no es un objeto de ordenador tangible o una memoria compartida en cualquier lugar, más bien es la suma del efecto de los ejecutables del tipo editor desplegados. La suma de los nodos desplegados es igual a la implementación de un solo DTD. Este solo DTD define el contexto en el cual cada autor trabaja. Debido a que un solo DTD virtual es cooperativamente utilizado, con la excepción del estilo, el contexto queda idéntico en cada sitio de edición. Dependiendo de sus tipos de capacidades, los autores cooperativamente expanden y aumentan el DTD virtual hasta que el área semántica sea considerada exhausta por los autores y otorgantes de privilegios. Cuando todos los privilegios de creación de contenido y contexto hayan sido revocados para todos los nodos, el documento distribuido virtualmente es considerado completo.

La presente invención implementa la prescripción del DTD para un grupo de escritores de documento, habilitando cada uno de ellos para producir simultáneamente cambios en el contenido en una base mutuamente exclusiva y exhaustiva con respecto al DTD todo el tiempo de visualización de texto de contenido y contexto de todo el documento simultáneamente. La invención conecta la estructura gramática subyacente invisible a los comportamientos de escritura de autores convencionales atando estilos de texto actualmente aceptados a la gramática en el momento cuando el documento ejecutable es creado. El ejemplo DTD etiqueta y por lo tanto nunca visible al autor o suscriptor, más bien el significado del contexto es re direccionado a través de las señales de estilos. Este proceso de estilo de contexto es análogo a lo que ocurre para documentos de lenguaje natural.

5.2 Tipos de Entidad

Una realización preferida de la presente invención comprende cinco tipos cada uno con diferentes capacidades. Los cinco tipos de entidad son (1) generador de contexto raíz, (2) autor de contexto raíz, (3) autor de contexto, y (4) autor de contenido y (5) suscriptor. Los tipos están causalmente relacionados. Cada tipo tiene capacidades, privilegios y comportamientos de código suplementarios que están predefinidos en el momento de la creación, pero que son modificables en el momento de la ejecución. Estos tipos de entidad están además clasificados en dos tipos en general: editor tipo y suscriptor tipo. El autor de contexto raíz, los autores de contexto y los autores de contenido son todos editores tipo. A todos los editores tipo se les asignan nodos, y todos los editores tipo son capaces de editar un contenido de documento. El autor de contexto raíz y los autores de contexto son capaces de crear y modificar así el contexto creando o cambiando los sub nodos. Un suscriptor tipo, a diferencia de todos los editores tipo, no se asigna a un nodo particular y es por lo tanto de tipo no editor. Los suscriptores son considerados terceros y así son solo capaces de presenciar la evolución del documento.

Para cada tema de nodo raíz la invención es realizada por una colección de editores y de no editores todos de los cuales son privados para la creación del documento.

50 5.3 Capacidades y Partes de Documento

5.3.1 El Documento

5.3.1.1 Inspección

Todos los tipos tienen la capacidad de ver las secciones de documento. Una sección es una representación visual del nodo del contexto y el contenido estilizado. Es posible con la invención para un nodo conceder a cualquier de todos los nodos descendentes el acceso a ninguno de los documentos o al documento entero, o cualquier porción en este.

5 5.3.1.2 Edición de Contexto y Contenido

10 Todos los editores tipo tienen la capacidad de editar el contenido de su nodo. El contenido se refiere al material de texto entre las marcas de inicio y fin de la instancia SGML. Allí hay uno y sólo un autor por nodo DTD quien puede editar el contenido. Con respecto a todas las porciones de contexto del documento, todos los tipos de entidad son suministrados con privilegios de solo lectura. La expresión estilizada del contexto siempre ocurre solo en el momento de creación del nodo. Consecuentemente, ningún nodo receptor puede alterar las señales de contexto del nodo recibido. También en lenguaje natural, el contexto existe antes del ejercicio de producir el contenido de lenguaje. Solo el autor del contexto raíz puede ajustar el estilo inicial de las porciones de representación de contexto del documento. Autores de contexto subordinados pueden modificar el estilo inicial solo dentro límites permitidos, y sus modificaciones están limitadas a sub nodos recientemente creados.

15 5.4 El Nodo

El nodo y sus relaciones semánticas con sus parientes y niños representan el contexto de lenguaje. El DTD mantiene esta relación directamente todo el tiempo permitiendo la representación para ser manejable por el ordenador.

20 Un nodo es la dirección de ordenador del autor y su elemento semántico identificable en el DTD. Un nodo comprende un código ejecutable que encarna las capacidades con las cuales fue creado. Sobre la creación de un nuevo nodo, cada nodo anterior y el suscriptor es informado que un nuevo nodo ha sido añadido al documento. Estos resultados informativos en adición a la sección de nuevos nodos dentro de todas las instancias del documento. La operación informativa modifica la operación de todos los destinatarios de forma que el nuevo nodo puede continuar el proceso informativo como ocurran cambios. Subsecuentemente los cambios al contenido del nuevo nodo afectan solamente el nuevo nodo correspondiente en un documento destinatario.

25

5.4.1 Creación de un Nuevo Nodo

5.4.1.1 Nodos Semánticos Hijos

30 El autor del contexto raíz y autores de contexto tienen la capacidad de crear y asignar nuevas porciones descendientes semánticas del documento a nuevos autores. La creación de nodo es el acto atómico de la semántica además de la diferenciación del creador del nodo, creando el código ejecutable para el nuevo nodo y reenviando el código al autor objetivo. El nuevo autor es concedido leyendo y escribiendo los privilegios para el contenido de ese nuevo nodo. Cada creación y asignación resulta ya sea en la creación de un autor de contexto o la creación de un autor de contenido. En cualquier momento durante la ejecución, los autores de contexto pueden crear nuevos autores de contexto y autores de contenido semánticamente subordinados. Los autores de contenido no tienen esta capacidad.

35

Cualquier nodo de autor o suscriptor tiene la capacidad innata de crear otro suscriptor, aunque este privilegio puede ser revocado en el momento de la ejecución o la capacidad simplemente no puede ser provista en el momento de creación del código. Un suscriptor no posee un nodo particular.

Esto es sólo posible para crear un nodo en el documento cuyo contenido es modificable por uno y solo un autor.

40 5.4.1.2 Administración y Propiedad del Nodo

Un autor puede poseer y operar más de un nodo. Él puede también administrar directamente cualquier sub nodo o alternativamente conceder o revocar privilegios para nodos subordinados para llevar a cabo algunas o todas de sus tareas.

45 Cualquier algoritmo de implementación empleando un código suplementario puede ser utilizado para efectuar el control de edición sobre ocurrencias en particular de grupos de contenido de múltiples ocurrencias ("grupos de contenido" es un término SGML) o pueden permitirse elementos como indicadores de ocurrencia SGML, que es, *, + y ?, el tiempo en el cual el algoritmo no compromete los derechos de propiedad del autor básico.

5.4.2 Pasos para Crear un Nuevo Nodo

En una realización preferida, estos son los cinco pasos involucrados en la creación de nuevos nodos.

5.4.2.1 Selección de un Sub Nodo Semántico

La selección de un sub nodo nuevo es el mismo proceso que ocurre cuando se selecciona un nuevo nodo utilizando un editor DTD.

5 5.4.2.2 Interfaz de Diseño de Usuario

Con el nuevo nodo llamado y ubicado apropiadamente en el árbol DTD, una interfaz de usuario para el nuevo nodo es creada. La interfaz de usuario habilitada los autores para orientarlos visualmente con respecto al contexto de todo el documento, y habilita los autores para ingresar ediciones de documento en el área de contenido asignada del documento.

10 Las señales estructurales visuales de la interfaz de usuario para el nuevo nodo pueden implementarse en el resto del documento. La creación de la interfaz de usuario implica la selección de señales estructurales visuales de selección dentro de los privilegios y parámetros heredados de nodos ancestrales. Este proceso de selección resulta en un código ejecutable en el objetivo de la fuente. Otros propietarios e información de conexión de red son recolectados.

15 En el momento de la creación para ambas partes del documento de contexto y contenido, señales visuales de usuario y preferencias para indicación al inicio y final de las partes del documento o la finalización de indicación de edición de cambios a una parte del documento, son extraídas desde el creador utilizando sesiones de diálogo estándar. Con respecto en completar el cambio del contenido, este proceso señalador es una parte de la estilización de la interfaz de usuario y sigue señales de escritura de documento normal que indican que el autor está dejando o entrando en un área de cambio. Desde que el cambio de contexto no ocurre en un momento de ejecución no hay acciones pero solo señales visuales que sirven para orientar al usuario con respecto a la estructura del documento.

Nuevos autores de contexto tienen la capacidad de diseñar interfaces de usuario para cualquiera de los sub nodos que este pueda crear. Nuevos autores de contenido no tienen esta capacidad.

5.4.2.3 Elección de Privilegios y Comportamientos Suplementarios

25 Más allá de las capacidades innatas del nodo tipo que ha sido creado, el creador del nodo puede conceder cualquier conjunto de privilegios o dotarlo con cualquiera de las capacidades que fueron heredadas en el momento de la creación. Este conjunto de privilegios y capacidades pueden ser alterados o revocados o modificados en el momento de la ejecución por sus nodos antecesores.

30 Nuevos autores de contexto tienen la capacidad de establecer privilegios y legar comportamientos a sub nodos que puedan ser creados. Nuevos autores de contenido no tienen esta capacidad.

5.4.2.4 Implementación del Nodo

35 Los nodos de autor de contexto son capaces de desplegar el nuevo código ejecutable al nuevo autor del ordenador. Además, los nodos de autor de contexto subsecuentemente creado contienen la capacidad de desplegar ejecutables que estos puedan crear a cualquiera de los nuevos sub nodos que puedan crear. Los autores de contenido no tienen esta capacidad.

5.4.2.5 Actualización de Autores y Suscriptores

El paso final es actualizar los autores actuales y suscriptores con la nueva sección en el documento que este nodo representa. Sin el reconocimiento que todos estos pasos han llevado a cabo con éxito, ninguno de los del nuevo código serán activados y revertidos al estado previo del documento que tenga lugar.

40 5.4.3 Capacidades Estándar Incluidas

5.4.3.1 Visualización de Cambios de Contexto y Contenido

Un nodo ejecutable recientemente añadido incluye por defecto la capacidad de recibir los cambios de contenido de cualquier y todos los otros nodos. Esto aplica a todos los editores tipos y suscriptores.

5.4.3.2 Multiplicación de Cambios de Contenido

Un nodo de autor recientemente creado incluye por defecto la capacidad de insertar cambios de contenido a la lista de todos los autores y suscriptores del nodo. Esto aplica a los editores tipo.

5.4.3.3 Aceptar las Nuevas Direcciones

5 Un nodo editor recientemente creado incluye por defecto la capacidad de aceptar y adicionar a su lista de distribución, subsecuentemente direcciones de autor añadidas a las cuales el nodo editor recientemente creado enviará los cambios. Esto aplica a los editores tipo.

5.5 Crecimiento de Documento Adicionales

10 La presente invención encarna dos incrementos separados pero simultáneos y rutas de cambio, que es paralelo a las formas de comportamiento del lenguaje natural. Un camino es simultáneo al cambio de contexto y el otro es simultáneo al cambio de contenido. El cambio en general abarca la creación, modificación, eliminación, concesión y revocación de caminos y características del documento. Los autores de contexto simultáneamente crean y alteran nuevos sub nodos, y los autores de contexto y contenido crean y alteran el contenido.

5.5.1 Generador de Contexto Raíz

15 En una realización preferida, el primer paso de la invención es crear un código de ordenador ejecutable para el autor de contexto raíz. Es seleccionado un tema para el nodo raíz, estilización del contexto es ajustado para parámetros por defecto, y un ejecutable es generado y desplegado al objetivo de autor de contexto raíz.

5.5.2 Autor de Contexto Raíz

20 En una realización preferida, solo hay un autor de contexto raíz por documento, quien supervisa y controla la creación del documento inicial. El control de cambio del DTD es eventualmente jerárquico distribuido a todos los autores de contexto, o este puede mantenerse centrado en el autor de contexto raíz, o el control puede ser jerárquicamente distribuido entre cualquiera de los grados. El autor de contexto raíz puede crear cualquier número de modo de nodo subordinado semánticamente y con privilegio ejecutable.

25 El autor de contexto raíz modifica el contexto de estilización por defecto e incluso si es deseado el nombre del nodo raíz. Cualquiera de los privilegios y comportamientos que son considerados necesarios para los autores para escribir exitosamente el documento distribuido son añadidos al ejecutable. Después de que es configurado para un sub nodo en particular durante el proceso de creación del sub nodo, es este conjunto de privilegios y comportamientos que los ejecutables del sub nodos son dotados de cuando son creados.

5.5.3 Autores de Contexto

30 En una realización preferida, el autor de contexto solo podrá configurar la estilización del contexto, los privilegios y los comportamientos del software para los sub nodos que este crea. Los autores de contexto pueden crear cualquier número de ejecutables de nodo de modo semántico y con privilegios subordinado.

Los autores de contexto pueden crear o editar el contenido cualquier número de veces.

5.4.4 Autores de contenido

Los autores de contenido pueden crear o editar contenido cualquier número de veces.

35 5.5.5 Suscriptores

Los suscriptores ven el documento en la medida que evoluciona.

5.6 Ilustración de Evolución del Documento

40 La FIG. 1 es una representación gráfica de la generación de un autor de contexto raíz en un momento cero. En esta instancia, el generador de contexto raíz (8) genera un autor de contexto raíz para empezar el proceso de generación del documento. (10) puntos a uno de 4 ejecutables de autores de contexto raíz se han generado.

5.6.1 Interacción del Nodo Ejecutable

La FIG. 2 es una vista jerárquica de la presente invención ilustrando una sola vez en esta ilustración la vida del documento. Para las formas de generalidades del bien han sido sustituidos por nodos con nombre. Un borde grueso del nodo denota un nodo de autor de contexto y un borde delgado un nodo de autor de contenido. Ya que los suscriptores no poseen nodos, ellos no han sido incluidos con el fin de reducir la complejidad.

- 5 En la FIG. 2 el nodo de contexto 10a ha creado el sub nodo del autor de contenido 12a y un sub nodo de autor de contexto 14a. Entre el momento cero y un momento, ha ocurrido dos cambios de contexto.

La FIG. 3 ilustra la FIG. 2 en un estado de ejecución con los nodos creados desplegados como ejecutables 10b, 12b y 14b. Cada autor ejecutable contiene los tres nodos. Han ocurrido tres cambios en el contenido desde el momento uno resultando en 6 mensajes de contenido a los otros ejecutables en sus sitios.

- 10 La FIG. 4 es una vista jerárquica de la ilustración de documento en el momento dos. Entre el momento uno y el momento dos, el autor de contexto 14b ha diferenciado además su nodo 14a en dos sub nodos de autor de contenido 16a y 18a.

- 15 La FIG. 5 ilustra la FIG. 4 en un estado de ejecución con los nodos creados desplegados como ejecutables 10b, 12b, 14b, 16b y 18b. Cada autor ejecutable ahora contiene todos los cinco nodos. Cada autor ha hecho un cambio de contenido desde el momento dos resultando en 20 mensajes de contenido a los otros ejecutables en sus sitios.

5.6.2 Generación del Nodo de Contexto Ejecutable

- 20 La FIG.6 es una vista esquemática de la presente invención para propósitos de ilustración. El autor de contexto raíz del Manual Merck (30) ya ha creado sub nodos Cardiovasculares (32) y Pulmonares (34). Nodos de Acercamiento al Paciente (48) y Pruebas Pulmonares (50) fueron creados ya sea por el Manual Merck (30) o Pulmonar (34). Cardiovascular (32) o el Manual Merck (30) ya han creado el Ejercicio del Corazón (40) y Ramas de la Aorta (42) y es en el proceso de ingreso de la configuración de la información en el cuadro de diálogo (54). Después de la terminación, es creado el ejecutable del Corazón del Pericardio (38).

- 25 Para propósitos de ilustración, el cuadro de diálogo (54) muestra tres piezas de información que son solicitadas. (56) pregunta por el texto literal que es para reemplazar la etiqueta de inicio del nodo. (58) pregunta por el nombre del autor y su dirección web (59) (por ejemplo, dirección IP, dirección de red).

- La FIG. 7 es un ejemplo de un documento del autor visto en el momento de ejecución. El nombre del nodo es Cardiovascular (32) cuya representación de contexto de solo lectura literal es "Enfermedades del Corazón y del Pericardio" (72) y cuyo contenido (66) es de lectura y escritura como es indicado por el cursor (68). Otras áreas de contenido, (60), (62) y (64) son solo de lectura ya que no son de propiedad de este nodo.

- 30 Cuatro contextos y contenido del nodo son visibles en la pantalla. Cinco señales de estilización son utilizadas para indicar de una forma convencional en el documento la estructura del DTD y con ella, el documento. La sangría indica la posición ancestral en el árbol DTD. Los decimales numéricos (74) de señal visual también muestran la jerarquía implícita. Letras en negrita, sombreado detrás del texto y privilegios de solo lectura trabajan en conjunto para indicar en cuales porciones del texto tienen los literales de etiquetas ((70), (72), (74) y (76)), que se asocian con la etiqueta del nodo en el momento de la creación. Privilegios de lectura y diferentes de sombreado de texto indican y dirigen al autor en cuanto a qué áreas no son sus áreas de contenido (60), (62) y (64). (70)'s "DESÓRDENES CARDIOVASCULARES" por lo tanto representan la etiqueta de instancia <ManualMerck>. El literal de etiqueta de finalización ha sido omitida lo cual es legal SGML ya que el comienzo de "1.1.1 Enfermedades del Corazón y del Pericardio" (72) indica la terminación del elemento del nodo (70).

- 40 La FIG. 8 (80) muestra la FIG. 7 después de que es exportada como una instancia SGML regular sin literales de etiquetas u otros artefactos de presentación.

5.7 Método para Producir una Compilación de Información Ordenada

El paso de inicio del método es crear un código de ordenador ejecutable que tenga la capacidad de crear autores de contexto raíz. En una realización preferida, hay un autor de contexto raíz por documento.

- 45 El código ejecutable del generador del contexto raíz tiene la capacidad de crear editores tipo de contexto raíz, donde dicho generador de contexto raíz tiene al menos las siguientes capacidades: (1) la capacidad de crear un código ejecutable asociado al nodo raíz para los autores de contexto raíz; y (2) la capacidad de distribuir dicho código ejecutable a través de un medio de comunicación para objetivos de autores de contexto raíz.

Este autor de contexto raíz es el nodo semántico más alto en el documento y es de editor tipo. El código de ordenador ejecutable subyacente al autor del contexto raíz incluye el código de ordenador que tiene al menos las siguientes capacidades: (3) la capacidad de mantener y actualizar una inscripción o lista de cada nodo adicionado al documento; (4) la capacidad de generar una interfaz inicial de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (5) la capacidad de propagar los cambios de contenido de su propio nodo a todos las otras vistas de nodos réplica en el documento distribuido; (6) la capacidad de crear un código ejecutable para nodos subordinados semánticamente, donde el nodo tipo es seleccionado de un grupo consistente de autor de contexto y autor de contenido; (7) la capacidad de distribuir el código resultante ejecutable a través de un medio de comunicación al autor etiquetado; (8) la capacidad de crear un código ejecutable para un suscriptor; (9) la capacidad de distribuir el código ejecutable del suscriptor resultante a través de un medio de comunicación al suscriptor etiquetado; (10) la capacidad de crear, configurar y supervisar las interfaces del nodo descendiente de usuario, los privilegios del nodo descendiente y el código de ordenador no suplementario descendiente; y (11) la capacidad de poblar el documento en el inicio. Nodos subordinados pueden heredar dichas capacidades que pueden ser concedidas y revocadas por nodos de orden superior en el momento de la creación y en el momento de la ejecución. La interfaz de usuario habilita al autor para ingresar contenidos editados en un área asignada del documento.

Otro paso es la creación de un código ejecutable para un autor de contexto. El código de ordenador ejecutable subyacente al autor de contexto incluye un código de ordenador que tiene al menos las siguientes capacidades: (12) la capacidad de mantener y actualizar una inscripción o lista de cada nodo añadido al documento; (13) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar cambios del contexto y del contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (14) la capacidad de propagar cambios de contenido de su propio nodo a todos las otra réplicas de vista de nodos en el documento distribuido; (15) la capacidad de responder a solicitudes administrativas; (16) la capacidad de crear un código ejecutable para nodos subordinados semánticamente, donde el nodo tipo es seleccionado de un grupo consistente de autor de contexto y autor de contenido; (17) la capacidad de distribuir el código ejecutable resultante a través de un medio de comunicación al autor etiquetado; (18) la capacidad de crear un código ejecutable para un suscriptor; (19) la capacidad de distribuir el código ejecutable del suscriptor resultante a través de un medio de comunicación para un suscriptor; (20) la capacidad de crear, configurar y supervisar interfaces de nodo de usuario descendientes, privilegios de nodo descendiente y código de ordenador suplementario de nodo descendiente; y(21) la capacidad de poblar el documento en el inicio. Nodos subordinados pueden heredar estas capacidades que pueden ser concedidas y revocadas por nodos de orden superior en el momento de la creación y en el momento de la ejecución. La interfaz de usuario habilita al autor para ingresar contenidos editados en un área asignada del documento.

Como se puede ver, el autor de contexto raíz y los autores de contexto comparten varias de las mismas características. Sin embargo, mientras un documento puede tener múltiples autores de contexto, en una realización preferida, un documento solo tiene un autor de contexto raíz. Además, en una realización preferida, los autores de contexto raíz crean sus propias interfaces de usuario iniciales y conjuntos de código suplementario mientras un autor de contexto puede a lo sumo solo modificar capacidades heredadas. Adicionalmente, en una realización preferida, los autores de contexto responden a comandos administrativos, mientras que los autores de contexto raíz residen en la parte superior de la jerarquía semántica, y así no responden a tales comandos administrativos.

En una realización preferida, la creación de un editor tipo subordinado por un autor de contexto o el autor de contexto raíz es el primer paso en un método para crear un cambio de contexto. Una vez un editor tipo subordinado ha sido creado, autores de editor tipo pueden llevar a cabo todos los otros pasos del método para crear cambios en el contexto en cualquier orden en cualquier momento.

Otro paso en el método de la invención es la creación del código de ordenador ejecutable para un autor de contenido. Los autores de contenido son siempre nodos de rama en la jerarquía semántica debido a que ellos no tienen la capacidad de crear nodos subordinados adicionales. Nodos de rama no tienen nodos hijos subordinados. El código de ordenador ejecutable subyacente al autor de contenido incluye un código de ordenador que tiene al menos las siguientes capacidades: (22) la capacidad de mantener y actualizar una lista de recordatorio para cada nodo adicionado al documento; (23) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (24) la capacidad de propagar cambios de contenido de su propio nodo a todos las otras réplicas de vista de nodos en el documento distribuido; (25) la capacidad de responder a solicitudes administrativas; (26) la capacidad de crear un código ejecutable para un suscriptor; (27) la capacidad de distribuir a través de un medio de comunicación el código ejecutable del suscriptor resultante al suscriptor objetivo; y (28) la capacidad de poblar el documento en el inicio. La interfaz de usuario habilita al autor para ingresar ediciones de contenido en un área asignada del documento.

Cualquier editor tipo puede llevar a cabo el paso de un método para crear y propagar cambios de contenido en cualquier momento y en cualquier orden.

Otro paso en el método de la invención es la creación de un código de ordenador ejecutable para suscriptores. Los suscriptores no están asociados con cualquier nodo y estos no tienen la capacidad de crear nodos subordinados adicionales. El código ejecutable de ordenador subyacente al suscriptor incluye un código de ordenador que tiene al menos las siguientes capacidades: (29) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento, (30) la capacidad de responder a solicitudes administrativas; y (31) la capacidad de poblar el documento en el inicio.

Cualquier participante del documento, incluyendo editores tipo o suscriptores quienes tienen privilegios adecuados, pueden llevar a cabo el paso del método de crear suscriptores en cualquier momento.

El paso en el método para poblar el documento debe necesariamente ser el primer acto de cualquier sitio de código ejecutable. Después de poblar un sitio de documento, se denomina que es consistente. Sólo el otro momento en que una réplica de documento no es consistente es durante el cambio de momentos de transmisión o si el flujo de trabajo suplementario del código tipo deliberadamente retrasa o re direcciona mensajes de cambio antes que lleguen a su destino. Un documento inactivo de esta invención es un documento que todos sus mensajes de cambio han sido entregados.

5.8 Sistema para Producir una Compilación Ordenada de la Información

El sistema comprende un código de ordenador que tiene la capacidad de crear autores de contexto raíz. En una realización preferida, solo hay un autor de contexto raíz por documento.

El código ejecutable para el generador de contexto raíz tiene la capacidad de crear editores tipo de contexto raíz, donde dicho generador de contexto raíz tiene al menos las siguientes capacidades: (1) la capacidad de crear un código asociado ejecutable de nodo raíz para autores de contexto raíz; y (2) la capacidad de distribuir dicho código ejecutable a través de un medio de comunicación para los autores objetivos del contexto raíz.

Este autor de contexto raíz es el nodo semántico más alto en el documento y es del editor tipo. El código de ordenador ejecutable subyacente al autor de contexto raíz incluye un código de ordenador que tiene al menos las siguientes capacidades: (3) la capacidad de mantener y actualizar un recordatorio de lista de cada nodo adicionado al documento; (4) la capacidad de generar una interfaz de usuario inicial que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los nodos actuales y eventuales en el documento; (5) la capacidad de propagar los cambios de contenido de su propio nodo a todas las réplicas de vista de nodos en el documento distribuido; (6) la capacidad de crear un código ejecutable para nodos subordinados semánticamente, donde el nodo tipo es seleccionado de un grupo consistente de un autor de contexto y un autor de contenido, (7) la capacidad de distribuir el código ejecutable resultante a través de un medio de comunicación al autor objetivo; (7) la capacidad de crear un código ejecutable para un suscriptor; (9) la capacidad de distribuir el código ejecutable del suscriptor resultante a través de un medio de comunicación al suscriptor objetivo; (10) la capacidad de crear, configurar y supervisar interfaces de nodo de usuario descendentes, privilegios de nodo descendentes y un código de ordenador de nodo descendiente suplementario; y (11) la capacidad de poblar el documento al inicio. Los nodos subordinados pueden heredar dichas capacidades que pueden ser concedidas o revocadas por nodos de orden superior en los momentos de creación y de ejecución. La interfaz de usuario habilita al autor para ingresar contenidos editados en un área asignada del documento.

En una realización preferida el sistema además comprende un código ejecutable para un autor de contexto. El código de ordenador ejecutable subyacente al autor de contexto incluye un código de ordenador que tiene al menos las siguientes capacidades: (12) la capacidad de mantener y actualizar un recordatorio una lista de cada nodo añadido al documento; (13) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (14) la capacidad de propagar los cambios de contenido de su propio nodo a todas las otras réplicas de vistas de nodo en el documento distribuido; (15) la capacidad de responder a solicitudes administrativas; (16) la capacidad de crear un código ejecutable para nodo subordinados semánticamente, donde el nodo tipo es seleccionado de un grupo consistente de un autor de contexto y un autor de contenido; (17) la capacidad de distribuir el código resultante ejecutable a través de un medio de comunicación al autor objetivo; (18) la capacidad de crear un código ejecutable para un suscriptor; (19) la capacidad de distribuir el código ejecutable del suscriptor resultante a través de un medio de comunicación al suscriptor objetivo; (20) la capacidad de crear, configurar y supervisar interfaces de usuario de nodo descendiente, privilegio de nodo descendiente y código de ordenador suplementario de nodo descendiente; y (21) la capacidad de poblar el documento en el inicio. Los nodos subordinados pueden heredar aquellas capacidades que pueden ser concedidas y revocadas por nodos de orden superiores en los momentos de creación y ejecución. La interfaz de usuario habilita al usuario para ingresar contenidos editados en un área asignada del documento.

5 Como se puede ver, el autor de contenido raíz y los autores de contexto comparten varias de las mismas características. Sin embargo, mientras un documento puede tener múltiples autores de contexto, en una realización preferida, un documento solo tiene un autor de contexto raíz. Además, en una realización preferida, los autores de contenido raíz crean sus propias interfaces de usuario y conjuntos de códigos suplementarios mientras un autor de contexto puede a lo sumo solo modificar capacidades heredadas. Adicionalmente, en una realización preferida, los autores de contexto responden a comandos administrativos, mientras que los autores de contexto raíz residen en la parte más alta de la jerarquía semántica, y así no responden a tales comandos administrativos.

10 En una realización preferida, el sistema crea un cambio de contexto creando primero un editor tipo subordinado por un autor de contexto o el autor de contexto raíz. Una vez un editor tipo subordinado ha sido creado, los autores del editor tipo crean un cambio de contexto en cualquier orden y en cualquier momento.

15 En una realización preferida el sistema es capaz de crear un código de ordenador ejecutable para un autor de contenido. Los autores de contenido son siempre nodos de rama en la jerarquía semántica debido a que estos no tienen la capacidad de crear nodos subordinados adicionales. Los nodos de rama no tienen nodos hijos subordinados a ellos. El código de ordenador ejecutable subyacente al autor de contenido incluye un código de ordenador que tiene al menos las siguientes capacidades: (22) la capacidad de mantener y actualizar un recordatorio lista de cada nodo añadido al documento; (23) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (24) la capacidad de propagar los cambios de contenido de su propio nodo a todas las otras réplicas de vistas de nodos en el documento distribuido; (25) la capacidad de responder a solicitudes administrativas; (26) la capacidad de crear un código ejecutable para un suscriptor; (27) la capacidad de distribuir a través de un medio de comunicación el código ejecutable del suscriptor resultante del suscriptor objetivo; y (28) la capacidad de poblar el documento en el inicio. La interfaz de usuario habilita al autor para ingresar un contenido editado en un área asignada del documento.

Cualquier editor tipo puede crear y propagar un cambio de contenido en cualquier momento y en cualquier orden.

25 En una realización preferida el sistema es capaz de crear un código ejecutable de ordenador para los suscriptores. Los suscriptores no están asociados con cualquier nodo y no tienen la capacidad de crear nodos subordinados adicionales. El código de ordenador ejecutable subyacente al suscriptor incluye un código de ordenador que tiene al menos las siguientes capacidades: (29) la capacidad de mostrar una interfaz de usuario que pueda recibir y mostrar los cambios de contexto y contenido de un documento de todos los otros nodos actuales y eventuales en el documento; (30) la capacidad de responder a solicitudes administrativas; y (31) la capacidad de poblar el documento al inicio.

Cualquier participante del documento, incluyendo editores tipo o suscriptores quienes tienen privilegios adecuados, pueden crear suscriptores en cualquier momento.

35 En una realización preferida del sistema, la población del documento debe necesariamente ser el primer acto de cualquier código ejecutable en el sitio. Después de popular un sitio del documento, se denomina que es consistente. La única vez que una réplica de documento no es consistente es durante el cambio en los momentos de transmisión o si el flujo de trabajo suplementario del código tipo deliberadamente retrasa o re direcciona los mensajes de cambio antes que lleguen a su destino. Un documento inactivo de esta invención es un documento todo cuyos mensajes de cambio han sido entregados.

40 Como es entendido por una persona con capacidades en la técnica, el propósito de la ilustración anterior es para clarificar la presente invención en lugar de limitar la presente invención.

REIVINDICACIONES

1. Un sistema para producir un documento distribuido (60, 62, 64, 66, 68, 70, 72, 74, 76) teniendo una compilación ordenada de información, el sistema comprende múltiples fragmentos de editor ejecutables (10b, 12b, 14b) que funcionan cooperativamente como una práctica, la declaración de documento tipo, cada uno de los múltiples fragmentos de editor ejecutables (10b, 12b, 14b) son un nodo específico de implementación de una práctica de declaración de documento tipo, donde dichos múltiples fragmentos de editor ejecutables están distribuidos entre múltiples sitios de una red de ordenador y repeticiones del documento completo residen en múltiples sitios de la red de ordenador, los múltiples fragmentos de editor ejecutables operan en un ambiente punto-a-punto sin un servidor central, el sistema permite múltiples autores (10a, 12a, 14a, 16a, 18a) para editar el documento distribuido y la declaración del documento tipo en forma simultánea, donde una edición hecha por uno de los múltiples autores (10a, 12a, 14a, 16a, 18a) es propagada entre las réplicas que residen en múltiples sitios de la red de ordenador, donde cada uno de los múltiples autores (10a, 12a, 14a, 16a, 18a) y suscriptores son permitidos para ver ediciones y cambios de declaración de documentos tipo hechas por otros de los múltiples autores (10a, 12a, 14a, 16a, 18a) en forma simultánea, donde cada autor de los múltiples autores (10a, 12a, 14a, 16a, 18a) le es asignado una entidad tipo estando asociada con los ejecutables correspondientes que definen el papel que un autor puede jugar en la creación de un documento distribuido (60, 62, 64, 66, 68, 70, 72, 74, 76), y donde solo es un solo autor por declaración de documento tipo quien puede editar el contenido y donde estos autores cooperativamente expanden y aumentan el área semántica en común, la declaración de documento tipo virtual, el sistema además tiene un control de cambio, el cual es jerárquicamente distribuido, y una autoridad en el cambio de contenido, el cual es mutuamente exclusivo y exhaustivo, el sistema además comprende:
- (a) un código de ordenador ejecutable para un autor generador de contexto raíz que comprende: (i) un código de ordenador para la creación de un nodo raíz asociado con un código ejecutable para al menos un autor de contexto raíz (10a) donde dicho autor de contexto raíz (10a) es un editor tipo; (ii) un código de ordenador para distribución a través de un medio de comunicación para dicho autor de contexto raíz (10a) dicho código ejecutable;
- (b) un código de ordenador ejecutable para dicho autor de contexto raíz (10a) que comprende: (iii) un código de ordenador para mantener y actualizar un recordatorio para cada nodo adicionado a un documento; (iv) un código de ordenador para la creación de una interfaz de usuario inicial donde dicha interfaz de usuario recibe y muestra el contenido de dicho documento de otros nodos y dicha interfaz habilita dicho autor de contexto raíz (10a) para ingresar contenidos editados en un área asignada de dicho documento; (v) un código de ordenador para propagar cambios de contenido de dicho autor de contexto raíz (10a) a todas las otras réplicas de vista de nodos en los sitios de autor y de suscriptor; (vi) un código de ordenador para la creación de un nodo raíz asociado a un código ejecutable para al menos un nodo objetivo donde el tipo de dicho nodo objetivo es seleccionado de un grupo consistente de un autor de contexto (14a) y un autor de contenido (12a) donde dicho autor de contexto (14a) puede administrar y es un editor tipo y dicho autor de contenido (12a) es un editor tipo, (vii) un código de ordenador para distribuir a través de un medio de comunicación a dicho nodo objetivo dicho nodo asociado a un código ejecutable; (viii) un código de ordenador para la creación de un código ejecutable suscriptor por al menos un suscriptor objetivo donde dicho suscriptor objetivo no es un editor tipo; (ix) un código de ordenador para distribuir a través de un medio de comunicación a dicho suscriptor objetivo dicho código ejecutable de suscriptor; (x) un código de ordenador para dicha administración donde dicha administración comprende la concesión en curso y la revocación de los privilegios del nodo descendente, la configuración en curso y las interfaces de usuario del nodo descendente, y la configuración en curso del código de ordenador del nodo descendente suplementario; (xi) el código de ordenador para poblar dicho documento al inicio;
- (c) un código ejecutable para dicho autor de contexto (14a) donde dicho autor de contexto (14a) comprende: (xii) un código de ordenador para mantener y actualizar un recordatorio de cada nodo añadido a un documento; (xiii) un código de ordenador para una interfaz de usuario donde dicha interfaz de usuario recibe y muestra el contenido de dicho documento y dicha interfaz habilita dicho autor de contexto (14a) para ingresar contenidos editados en el área asignada de dicho documento; (xiv) un código de ordenador para propagar los cambios de contenido de su propio nodo a todas las otras réplicas de vista de nodos en los sitios del autor y del suscriptor; (xv) un código de ordenador para responder a una solicitud administrativa; (xvi) un código de ordenador para la creación de un nodo asociado a un código ejecutable para al menos un nodo objetivo donde dicho nodo objetivo es seleccionado de un grupo consistente de un autor de contexto (14a) y un autor de contenido (12a) donde dicho autor de contexto (14a) puede administrar y es un editor tipo y dicho autor de contenido (12a) es un editor tipo; (xvii) un código de ordenador para distribuir a través de un medio de comunicación a dicho nodo objetivo dicho código ejecutable asociado; (xviii) un código de ordenador para la creación de un código ejecutable de suscriptor para al menos un suscriptor objetivo donde dicho suscriptor objetivo no es un editor tipo; (xix) un código de ordenador para distribuir a través de un medio de comunicación a dicho suscriptor objetivo dicho código ejecutable de suscriptor; (xx) un código de ordenador para dicha administración donde dicha administración comprende la concesión en curso y la revocación de los privilegios de nodo descendentes, la configuración en curso de las interfaces de nodo descendentes, y la configuración en curso del código de ordenador suplementario de nodo descendente; (xxi) un código de ordenador para poblar dicho documento al inicio;

- (d) un código ejecutable para dicho autor de contenido (12a) donde dicho autor de contenido (12a) comprende: (xxii) un código de ordenador para mantener y actualizar un recordatorio de cada nodo añadido a un documento; (xxiii) un código de ordenador para una interfaz de usuario donde dicha interfaz de usuario recibe y muestra el contenido de dicho documento y dicha interfaz capacidad dicho autor de contenido (12a) para ingresar contenidos editados en el área asignada de dicho documento; (xxiv) un código de ordenador para propagar cambios de contenido de su propio nodo a todas las otras réplicas de vistas de nodo en los sitios de autor y suscriptor; (xxv) un código de ordenador para responder a una solicitud administrativa; (xxvi) un código de ordenador para la creación de un código de suscriptor ejecutable para al menos un suscriptor objetivo donde dicho suscriptor objetivo no es un editor tipo; (xxvii) un código de ordenador para distribuir a través de un medio de comunicación a dicho suscriptor objetivo dicho código ejecutable de suscriptor; (xxviii) un código de ordenador para poblar dicho documento en el inicio;
- (e) un código ejecutable para dicho suscriptor donde dicho suscriptor comprende: (xxix) un código de ordenador para una interfaz de usuario donde dicha interfaz de usuario recibe y muestra el contenido de dicho documento; (xxx) un código de ordenador para responder a una solicitud administrativa; (xxxi) un código de ordenador para poblar dicho documento en el inicio.
- 15 2. El sistema de la reivindicación 1, donde la edición hecha por cualquier de los múltiples autores (10a, 12a, 14a, 16a, 18a) es inmediatamente propagada entre las réplicas.
3. El sistema de la reivindicación 1, donde la edición hecha por cualquier de los múltiples autores (10a, 12a, 14a, 16a, 18a) es propagada como una transacción atómica entre las réplicas.
- 20 4. El sistema de la reivindicación 1, donde las réplicas residen en la memoria de ordenador en múltiples sitios de la red de ordenadores.
5. El sistema de la reivindicación 1, donde las réplicas son persistidas escribiendo en discos duros de ordenador en múltiples sitios de la red de ordenadores.
6. El sistema de la reivindicación 1, donde la compilación ordenada es un documento SGML.
- 25 7. El sistema de la reivindicación 6, donde el documento SGML es seleccionado de un grupo consistente de XML y HTML.
8. El sistema de la reivindicación 1, donde los nodos producidos por dicho autor de contexto raíz (10a) y dicho autor de contexto (14a) son elementos válidos semánticamente dentro de una declaración de documento tipo.
9. Un método para producir el sistema de la reivindicación 1, el método comprende:
- (a) crear el código de ordenador ejecutable para el autor generador de contexto raíz (10a);
- 30 (b) crear el código de ordenador ejecutable para dicho autor de contexto raíz (10a);
- (c) crear el código ejecutable para dicho autor de contexto (14a);
- (d) crear el código ejecutable para dicho autor de contenido (12a);
- (e) crear el código ejecutable para dicho suscriptor.
10. El método como en la reivindicación 9, donde la compilación ordenada es un documento SGML.
- 35 11. El método como en la reivindicación 10, donde el documento SGML es seleccionado de un grupo consistente de XML y HTML.
12. El método como en la reivindicación 11, donde los nodos producidos por dicho autor de contexto raíz (10a) y dicho autor de contexto (14a) son elemento válidos semánticamente dentro de una declaración de documento tipo.

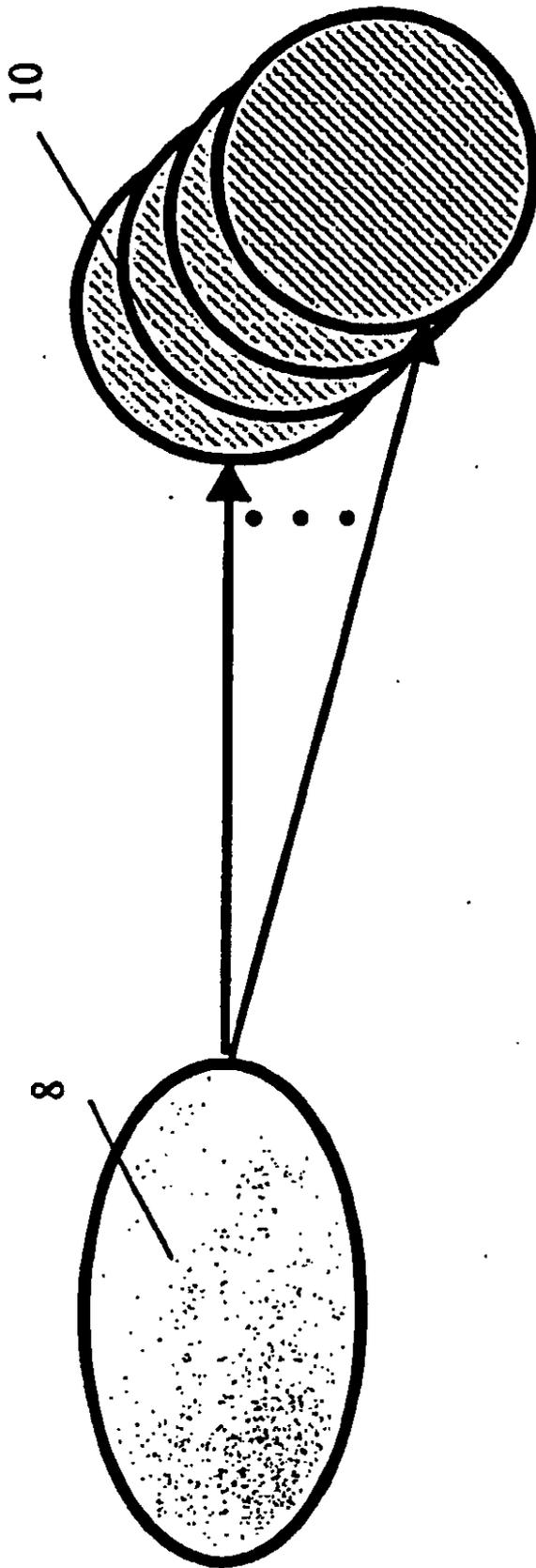


FIG.1

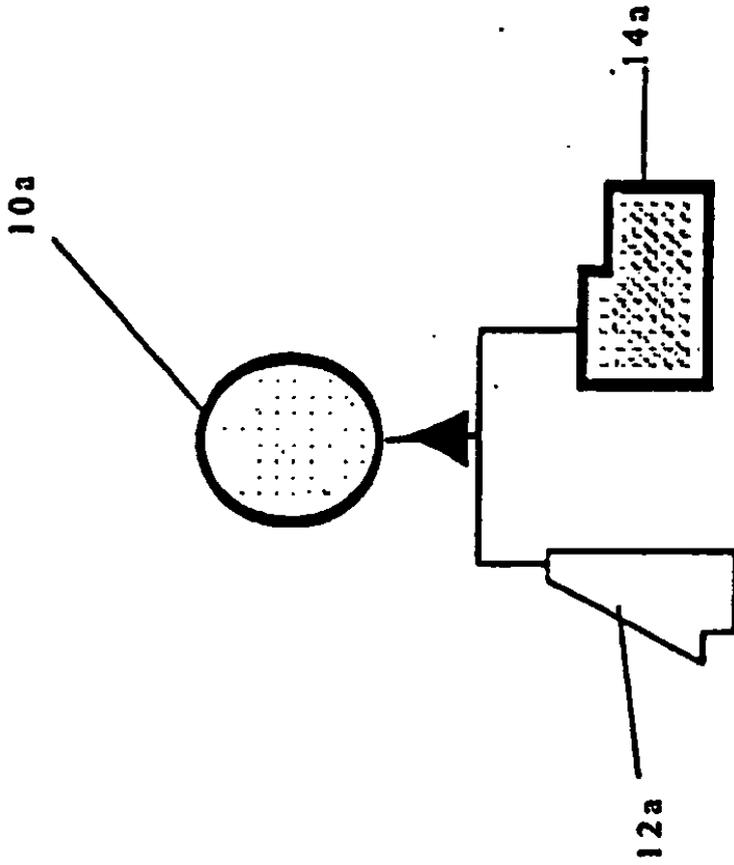


FIG.2

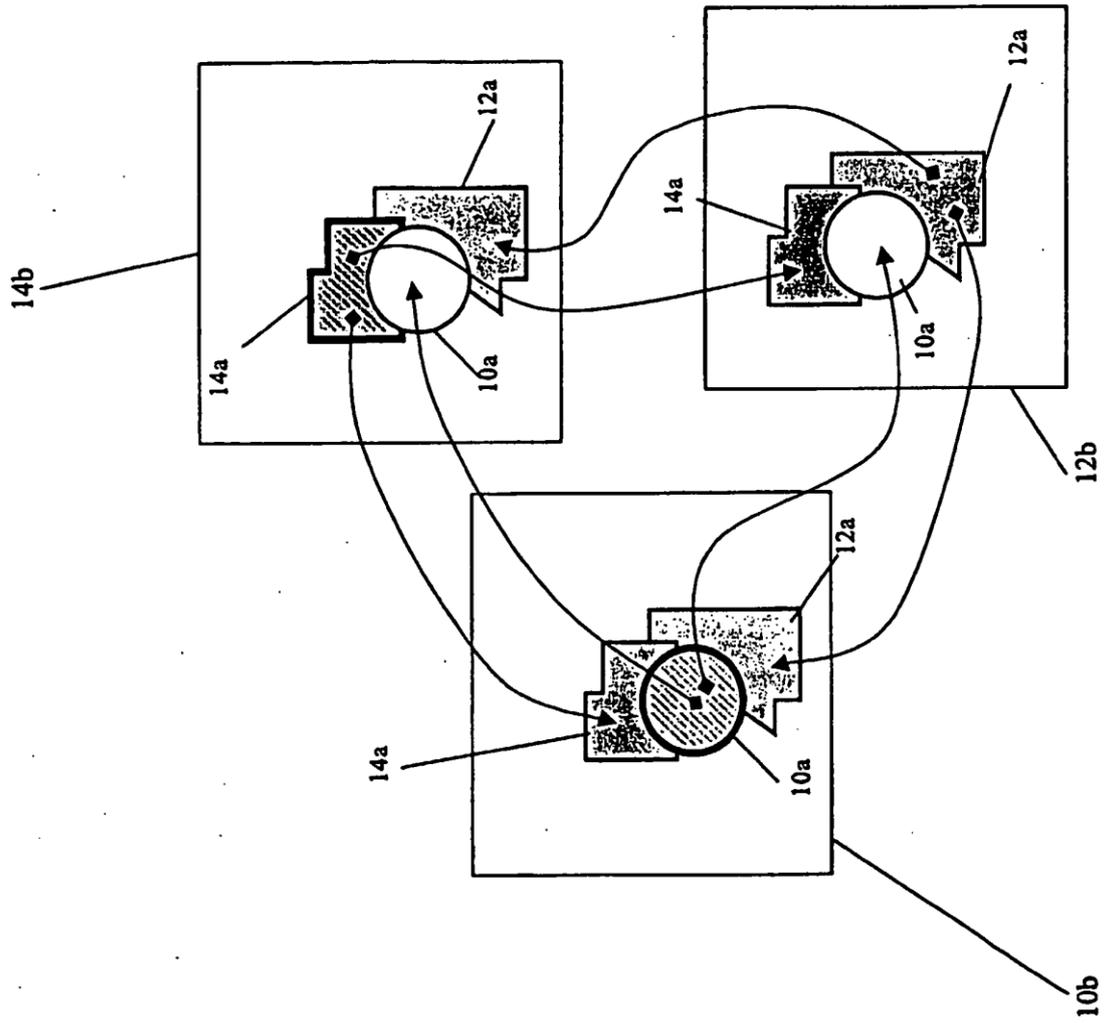


FIG.3

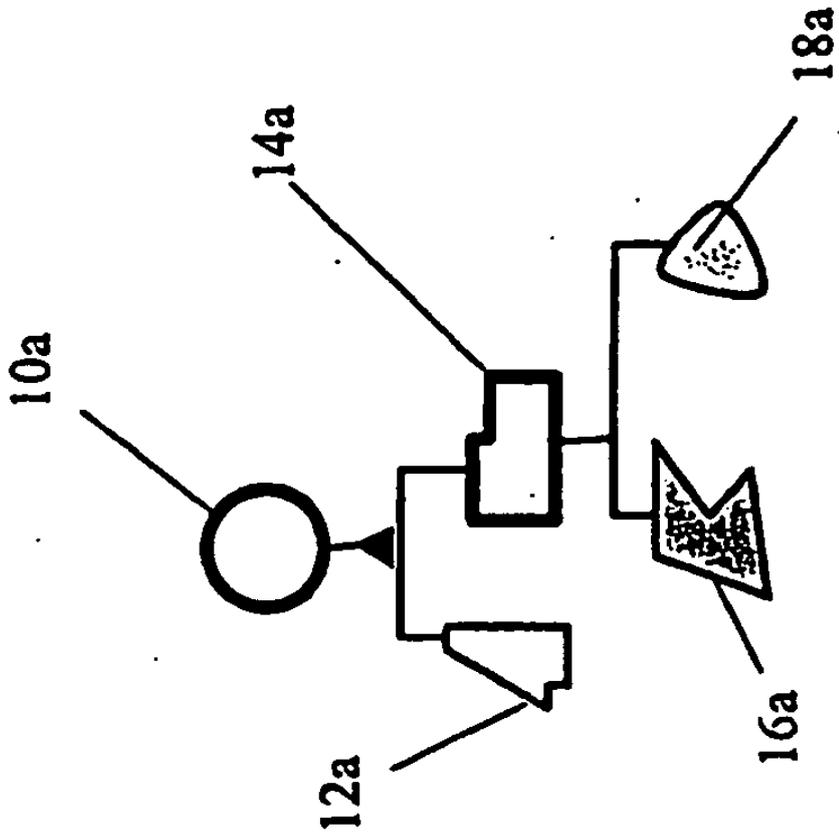


FIG.4

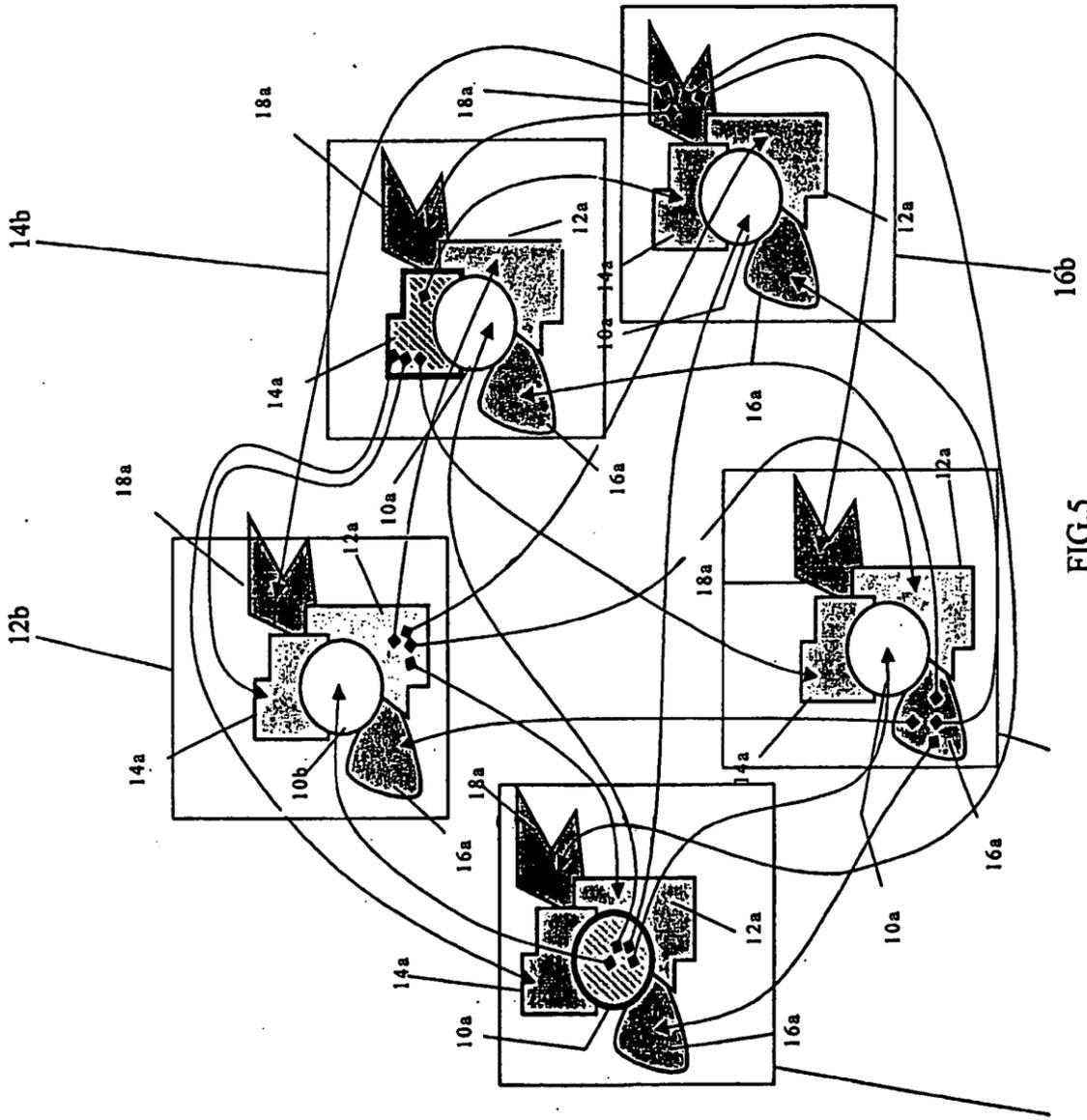


FIG. 5

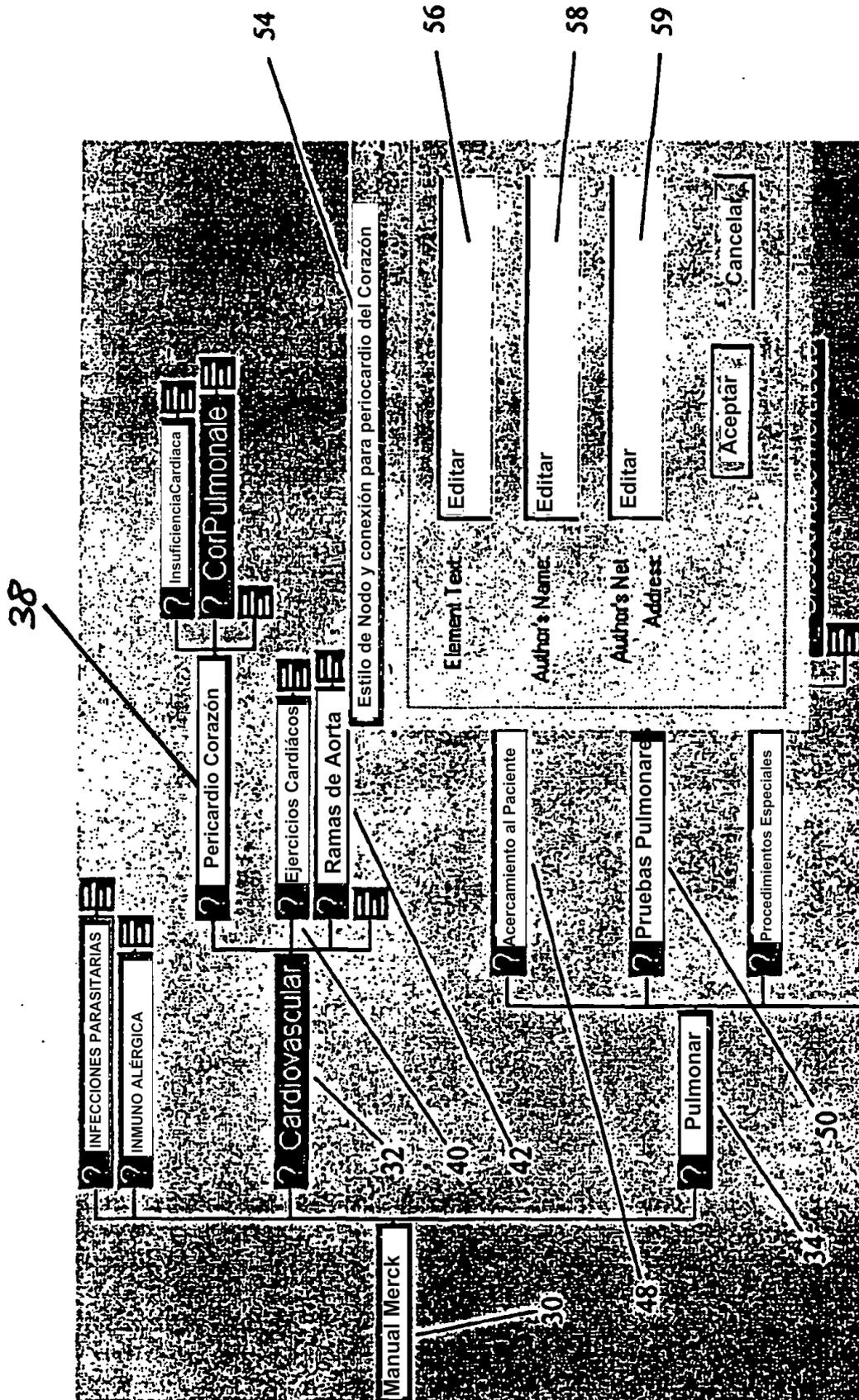


FIG. 6

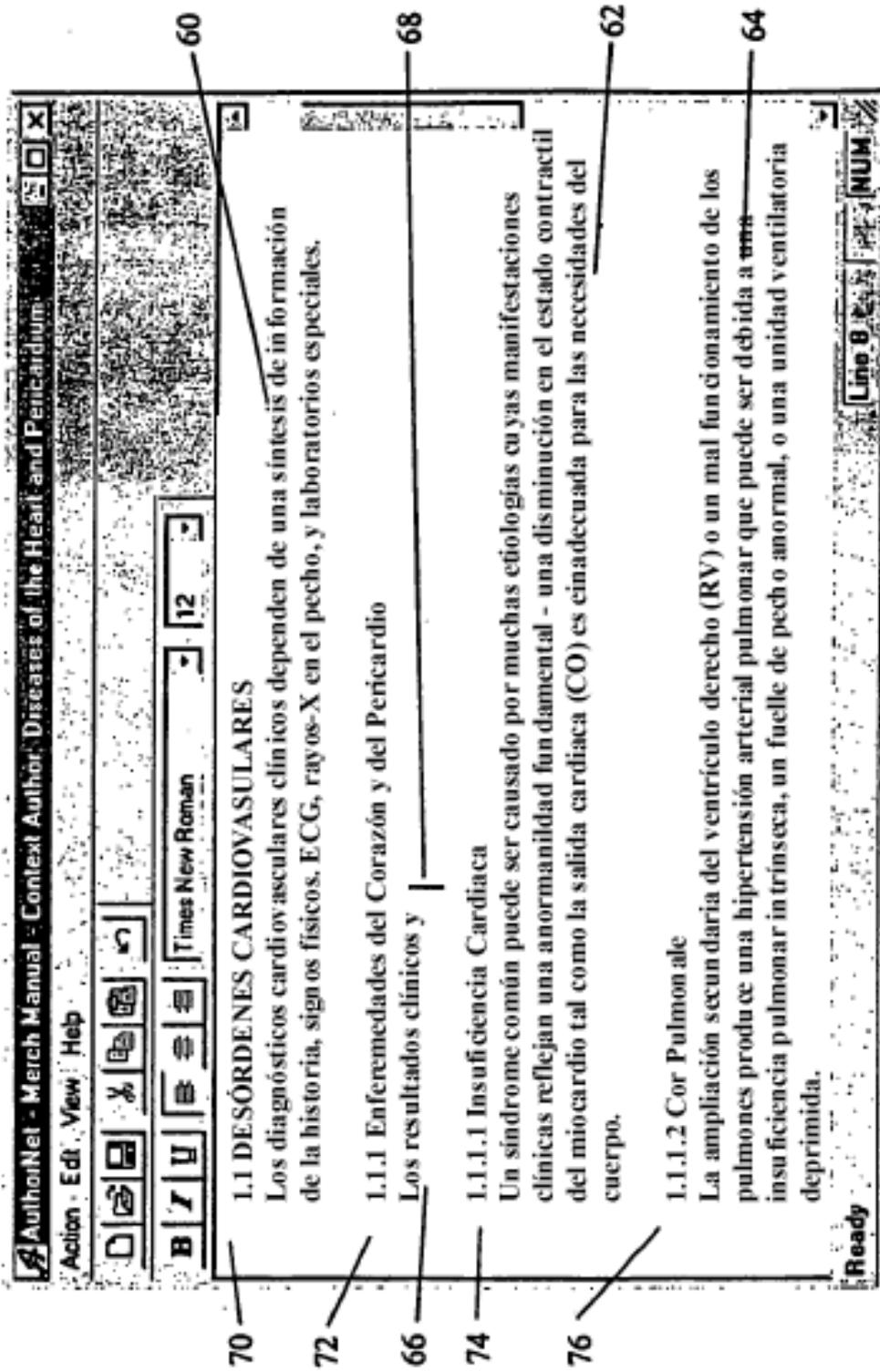


FIG. 7

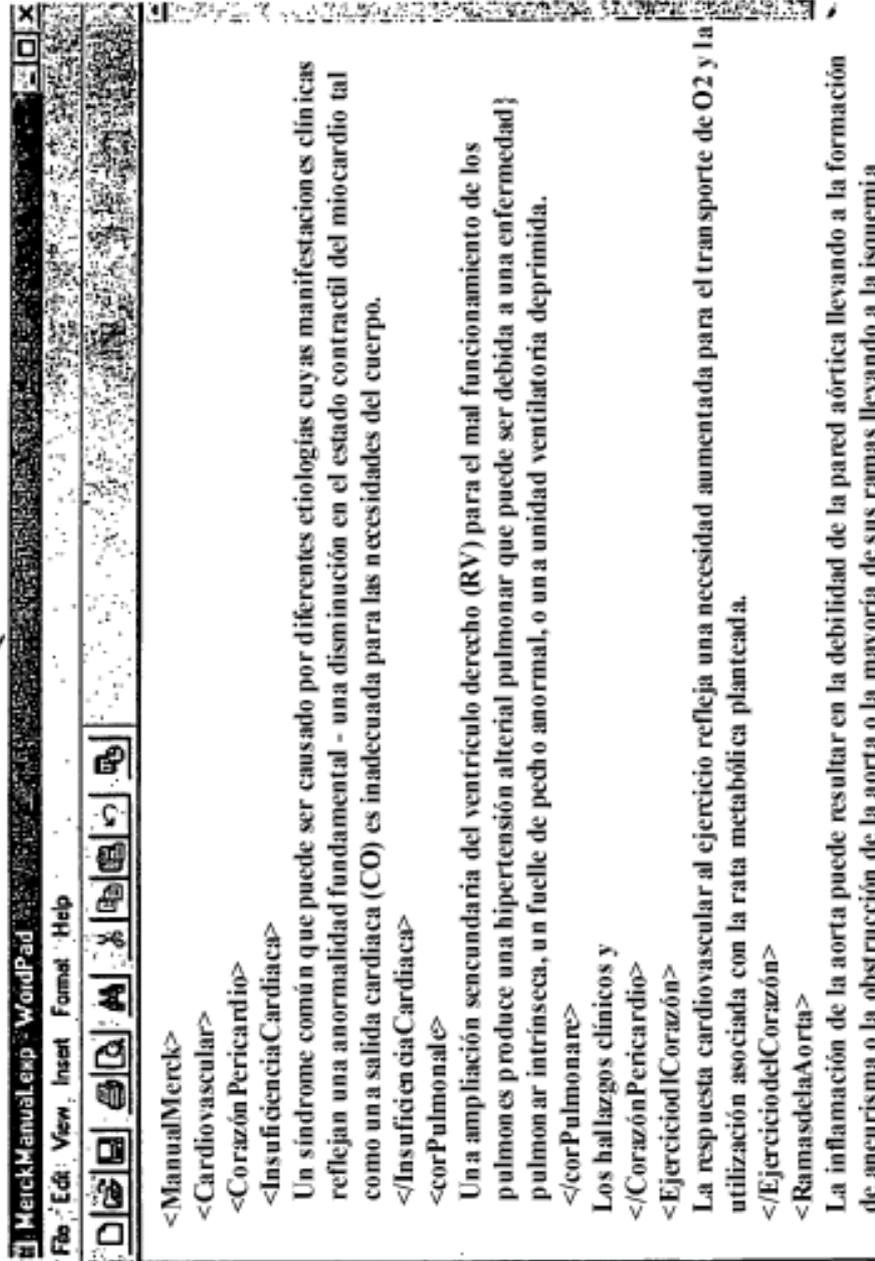


FIG. 8