

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 431 307**

51 Int. Cl.:

**H04N 21/8543** (2011.01)

**H04N 21/6543** (2011.01)

**H04N 21/81** (2011.01)

**H04N 21/236** (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **21.04.2003 E 03728451 (0)**

97 Fecha y número de publicación de la concesión europea: **17.07.2013 EP 1522191**

54 Título: **Soporte de funcionalidad de televisión común interactiva a través de la presentación de sintaxis de motor**

30 Prioridad:

**19.04.2002 US 373883 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**25.11.2013**

73 Titular/es:

**OPEN TV, INC. (100.0%)  
275 SACRAMENTO STREET  
SAN FRANCISCO, CA 94111, US**

72 Inventor/es:

**DELPUCH, ALAIN;  
WHITLEDGE, JAMES;  
MENAND, JEAN-RENE;  
BARBIER, EMMANUEL;  
HAUSMAN, KEVIN;  
HENSGEN, DEBRA y  
SU, DONGMIN**

74 Agente/Representante:

**TOMAS GIL, Tesifonte Enrique**

ES 2 431 307 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**

Soporte de funcionalidad de televisión común interactiva a través de la presentación de sintaxis de motor

5

Antecedentes de la invención

Campo de la invención

10

[0001] La invención se refiere generalmente a sistemas de televisión interactivos y más particularmente a un sistema y método para crear y controlar contenido de televisión interactiva.

Descripción de técnica relacionada

15

[0002] Los sistemas de televisión interactiva proporcionan unos medios para distribuir contenido interactivo además de audio y vídeo de televisión regular a un gran número de abonados. Los programas emitidos por estos sistemas pueden incorporar audio y vídeo televisivo, imágenes fijas, texto, gráficos interactivos y aplicaciones, y muchos otros componentes. También pueden proporcionar varios servicios, tal como el comercio a través de la televisión, guías de programa electrónico (EPGs), vídeos a pedido, y otras aplicaciones interactivas para espectadores. El contenido interactivo de la señal de televisión interactiva puede por lo tanto incluir código de aplicación, datos asociados al audio y vídeo, señales de control, datos brutos y muchos otros tipos de información. Esta información se puede combinar en una única señal o señales diferentes para transmisión a un receptor conectado a la televisión del espectador o el proveedor puede incluir sólo un subconjunto de la información.

20

25

[0003] La funcionalidad interactiva de la televisión generalmente está controlado por un receptor/decodificador integrado (IRD) o mecanismo similar, frecuentemente incorporado en un descodificador, conectado a la televisión. El IRD recibe la señal proporcionada por un proveedor de servicio de emisión u operador de sistema y separa la parte interactiva de la parte de audio-vídeo. El IRD usa la información interactiva para, por ejemplo, ejecutar una aplicación mientras la información de audio-vídeo se transmite a la televisión. El IRD puede combinar la información de audio-vídeo con gráficos interactivos o audio generados por la aplicación interactiva antes de transmitir la información a la televisión.

30

[0004] Los contenidos interactivos tales como códigos de aplicación o información acerca de programas televisivos pueden ser emitidos en un formato cíclico o repetitivo. Las piezas de información que son emitidas de esta manera forman lo que se puede llamar un "carrusel." Un carrusel puede incluir múltiples módulos de información, incluyendo un módulo de directorio que indica los módulos particulares que corresponden a una aplicación dada. Frecuentemente, un único carrusel se transporta como un flujo de datos contiguo. No obstante, es también posible multiplexar dos o varios carruseles en una sola transmisión de datos. Como alternativa en lugar de usar un formato de carrusel, algunos sistemas pueden utilizar un canal de retorno para solicitar y/o recibir contenido interactivo.

35

40

[0005] Los sistemas de emisión pueden transmitir información en un formato de carrusel para permitir a los receptores en el sistema obtener selectivamente piezas particulares de información en el carrusel sin requerir un canal de retorno de los receptores al servidor. Si un receptor en particular necesita una pieza en particular de información, puede simplemente esperar hasta la próxima vez que la pieza de información sea emitida de nuevo y luego extraer la información de la transmisión de datos emitida. Empleando carruseles para emitir información, el sistema puede eliminar la necesidad de conectar cada uno de los receptores con un servidor y además eliminar la necesidad del servidor de procesar peticiones individuales de información.

45

[0006] Las piezas de información, u objetos de datos, en un carrusel pueden ser destinados a combinarse en un único objeto de transmisión de datos para formar un programa. Este programa también puede contener datos de transmisión tales como audio o vídeo. Por ejemplo, un programa de concurso de televisión interactiva puede combinar audio y vídeo televisivo con contenido interactivo tales como códigos de aplicación que permite a los usuarios responder preguntas. Otro ejemplo sería un programa de noticias que combina audio y vídeo con código de aplicación que inserta precios de acciones de bolsa en un banner en la parte baja de la pantalla. Típicamente, cada programa se asocia con un canal correspondiente y, cuando el receptor de televisión interactiva selecciona un programa en particular, la información que se está transmitiendo en ese canal se descarga y el programa comienza.

50

55

[0007] Como los receptores televisivos se hacen más sofisticados, e incluyen la capacidad para acceder una gama mas amplia de datos y recursos, se han hecho esfuerzos para desarrollar mecanismos para el manejo de estos recursos adicionales. Por ejemplo, la especificación DVB MHP 1.1 y la especificación DAVIC 1.4.1 parte 9 definen un esquema URL para acceder a servicios de emisión. Puesto que la cadena de emisión DAVIC lleva servicio de información (SI) que contiene parámetros globales únicos para la localización de los servicios en una red de emisión, su esquema URL es capaz de direccionar servicios de una manera independiente de la red física.

60

[0008] Desafortunadamente, tales esquemas pueden no funcionar en redes ATSC u otras redes que definen formatos de señalización diferentes u homogéneos al propietario. Por lo tanto, se necesita un esquema nuevo más flexible. US 5,925,100 revela un sistema cliente/servidor que incorpora métodos para administrar la disponibilidad de objetos a

65

través de objetos semánticos "load sets". Un "load set" particular se asocia a cada objeto que puede ser solicitado por un cliente. Los métodos descritos son para administrar la recogida y descarte de objetos en base a objeto a objeto, no en base a página a página. Cada "objeto semántico" se empaqueta en un "almacenable", que incorpora listas de dependencia que indican el contexto en el que el objeto debe ser usado (es decir, con qué objetos dependientes).  
 5 Adicionalmente, un programador puede especificar que el conjunto de comportamientos se define a tiempo de ejecución, usando antiguas "precargas" proporcionadas por el sistema. Esto permite una aplicación de ejecución de objetos precargados basados en la dinámica de entonces existente del sistema. Con esta aproximación, la disponibilidad de un objeto en un entorno de objeto distribuido (por ejemplo; Internet) se mejora.

US 2001/001160 revela un sistema de entretenimiento interactivo que permite la presentación de contenido interactivo adicional junto a programas tradicionales de emisión de vídeo, tales como programas de televisión y películas. El contenido adicional se suministra como parte de la señal del mismo programa sobre la red de emisión, o separadamente sobre otra red de emisión. Una unidad de computación de espectadores se ubica en el hogar del espectador para presentar el programa y contenido adicional a un espectador. Cuando el espectador sintoniza a un canal en particular, la unidad de computación de espectadores consulta una guía de programación electrónica (EPG) para determinar si el programa actual transmitido en el canal es interactivo. Si lo es, la unidad de computación de espectadores ejecuta un navegador. El navegador usa una especificación de objetivos almacenada en el EPG para activar un recurso objetivo que tiene el contenido adicional para mejorar el programa emitido. El recurso objetivo contiene instrucciones de disposición de pantalla dando la orden de como el contenido adicional y el programa de contenido de vídeo deben aparecer uno en relación con el otro cuando se visualicen. Cuando los datos son descargados del recurso objetivo, la unidad de computación de espectadores responde a las instrucciones de disposición obtenidas del recurso objetivo para mostrar el contenido adicional al mismo tiempo que el programa de contenido de vídeo.

WO 01/26369 revela un activador de televisión interactiva que tiene un valor de atributo temporal que indica un momento futuro de cuando el activador debe ser ejecutado. En situaciones donde el activador no se puede enviar a la unidad receptora en el tiempo deseado de ejecución del activador (por ejemplo, debido a la limitación de la anchura de banda disponible de un canal de comunicación a la unidad receptora), el activador es enviado antes del momento futuro. La unidad receptora entonces ejecuta el activador en la unidad receptora en el momento futuro como indicado por el atributo temporal. El activador se puede enviar bastante antes del momento futuro de modo que una unidad de recepción que recibe el activador puede precargar un recurso de información identificado por el activador antes del momento futuro de manera que el recurso de información ya está disponible en la unidad receptora cuando el activador es ejecutado más tarde en el momento futuro. Activadores múltiples que identifican el mismo recurso de información pueden enviarse de modo que una unidad receptora que no reciba un activador enviado bastante antes de un momento futuro (un activador destinado a causar una precarga del recurso de información) recibirá otro activador al mismo recurso de información durante o cerca del momento futuro definido.

### 35 Resumen de la invención

[0009] Según un aspecto de la invención, se proporciona un método que comprende: recepción de una o más instrucciones, donde dichas instrucciones son indicativas de una presentación audio vídeo y/o gráfica que requiere un conjunto de recursos; determinando si dichas una o más instrucciones incluyen una instrucción prerequisite que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerequisite para iniciar la presentación; iniciar dicha presentación antes de la recepción todos los conjuntos de recursos mencionados, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerequisite; y, en respuesta a la determinación de que una o más instrucciones incluyen dicha instrucción prerequisite (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos y prohibir el inicio de dicha presentación hasta que sea adquirido dicho subconjunto de recursos.  
 40  
 45

[0010] Según otro aspecto de la invención, se proporciona un sistema de televisión interactivo que comprende: un servidor proxy remoto configurado para: recibir una o más instrucciones, donde dichas instrucciones son indicativas de una presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos; determinar si dichas una o varias instrucciones incluyen una instrucción prerequisite que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerequisite para iniciar la presentación; transmitir una o más primeras señales que identifican dicho subconjunto de recursos a un dispositivo de cliente remoto, en respuesta a determinar que una o más instrucciones incluyen dicha instrucción prerequisite; y transmitir una o más señales segundas que indican si dichas una o más instrucciones incluyen dicha instrucción prerequisite; un dispositivo de cliente configurado para: recepción de dichas primeras señales; recepción de dichas segundas señales; e iniciar dicha presentación antes de la recepción de todos los dichos conjuntos de recursos, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerequisite; y, en respuesta a determinar que una o más instrucciones incluyen dicha instrucción prerequisite (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos y prohibir el inicio de dicha presentación hasta que dicho subconjunto de recursos sea adquirido.  
 50  
 55  
 60

[0011] Según otro aspecto de la invención, se proporciona un dispositivo de cliente en un sistema de televisión interactiva, comprendiendo dicho dispositivo: un receptor configurado para recibir señales correspondientes a instrucciones que son indicativas de una presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos; y una unidad de procesamiento acoplada a dicho receptor, donde dicha unidad de procesamiento está configurada para: determinar si dichas una o varias instrucciones incluyen una instrucción prerequisite que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerequisite para el inicio de la presentación; iniciar dicha  
 65

presentación antes de la recepción de todos los dichos conjuntos de recursos, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerequisite; y, en respuesta a determinar que una o más instrucciones incluyen dicha instrucción prerequisite (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos y prohibir el inicio de dicha presentación hasta que el dicho subconjunto de recursos sea adquirido.

5 [0012] Según otro aspecto de la invención, se proporciona un medio portador que comprende instrucciones de programa ejecutable para: recibir instrucciones que son indicativas de una presentación audio, vídeo y/o gráfica que requieren un conjunto de recursos; determinar si dichas una o más instrucciones incluyen una instrucción prerequisite que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerequisite para la presentación; iniciar dicha presentación antes de la recepción de todos los dichos conjuntos de recursos, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerequisite; y, en respuesta a determinar que una o más instrucciones incluyen dicha instrucción prerequisite (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos e iniciar la prohibición de dicha presentación hasta que dicho subconjunto de recursos sea adquirida.

15 [0013] Se describe un método y mecanismo que permite a los autores de contenido usar instrucciones, tales como HTML, lenguajes de escritura, u otros lenguajes, con extensiones televisivas para crear y/o controlar contenido de televisión interactiva. El método y mecanismo se puede utilizar con programas digitalmente grabados al igual que con emisiones en directo.

20 [0014] En una forma de realización, un dispositivo en un sistema de televisión interactiva se configura para recibir una o más instrucciones proporcionadas por un autor de contenido que describe o indica de otra manera una presentación de audio y/o de vídeo. Incluido entre estas instrucciones están una o más instrucciones que indican que un subconjunto particular de recursos requerido para la presentación se consideran prerequisites. En respuesta a la detección de estas instrucciones, el suministro de la presentación se detiene hasta que los recursos de prerequisite sean obtenidos. En una forma de realización, las instrucciones se reciben por un servidor proxy ubicado centralmente que se puede configurar para recibir, transcodificar y transmitir contenido basado en web transcodificado a dispositivos de cliente. En cuanto se detectan instrucciones que indican recursos de prerequisite para una presentación, el servidor proxy transmite separadamente a las señales de dispositivos de cliente, o alguna otra indicación, que estos recursos son prerequisites. Como respuesta, el dispositivo de cliente que recibe las señales transmitidas puede tomar acciones para precargar estos recursos.

Breve descripción de las figuras

[0015]

- Fig. 1 es un diagrama de una forma de realización de un sistema de emisión de televisión.
- Fig. 2 es un diagrama de una forma de realización de una cabecera de red.
- Fig. 3 es un diagrama de bloques de una forma de realización de un dispositivo de cliente.
- Fig. 4 es un diagrama de una forma de realización de un sistema televisivo.
- Fig. 5 ilustra una forma de realización de un método usando instrucciones de prerequisite.

Descripción detallada

0. Resumen de sistema

45 [0016] En referencia a Fig. 1, se muestra una forma de realización de un sistema televisivo 100. En la forma de realización mostrada, se acoplan dispositivos de recepción 30 a diferentes fuentes de programación y/o contenido interactivos. Cada uno de los dispositivos de recepción 30 pueden comprender cualquier dispositivo adecuado, tal como un descodificador de sobremesa(STB), una televisión (TV), un grabador de vídeo (VCR), un aparato de vídeo digital (DVR), unos asistentes digitales personales (PDA), un ordenador personal (PC), una consola de videojuegos, o un teléfono móvil/celular.

55 [0017] Se incluye en la forma de realización de Fig. 1 una emisora de radioemisión 16 acoplada a receptor(es) 30 a través de un medio de transmisión 17 y backchannel 26. Además, se acoplan receptor(es) 30 a una fuente 18 y a una fuente 19 a través de una red 20. Además, se acopla la emisora de radioemisión 16 a una fuente remota 13, e Internet 60. En la forma de realización mostrada, la emisora de radioemisión 16 incluye fuentes 14 y 15 y transmisor 22. El medio de transmisión 17 puede comprender un sistema 23 con base de satélite, un sistema 24 con base de cable, un sistema 25 con base de un servicio de distribución terrestre o multipunto múltiple (MMDS), una combinación de estos sistemas, o algún otro sistema de transmisión adecuado.

60 [0018] En la forma de realización de Fig. 1, la estación de emisión 16 puede incluir una variedad de fuentes de contenido 14, 15, y 60 para ser utilizadas y transmitidas por el transmisor 22. Las fuentes de contenido 14 y 15 pueden incluir bases de datos, servidores de aplicación, otras fuentes de audio/vídeo, u otras fuentes de datos. En una forma de realización, se puede crear contenido en una fuente 14 que puede incluir una estación de autoría configurada para crear tal contenido. Una estación de autoría puede incluir un terminal de trabajo informático configurado con software que ayuda en el desarrollo de contenido interactivo. Una estación de autoría puede ser parte de emisora de radioemisión 16

en cuyo caso el transporte del contenido creado puede ser a través de una red de computación local, o configuración similar. Alternativamente, una estación de autoría puede ser ubicado remotamente 13 desde la emisora de radioemisión 16. En una forma de realización donde la estación de autor no está directamente acoplada a emisora de radioemisión 16, el contenido creado por una fuente 13 se puede transportar a la emisora de radioemisión 16 a través de Internet, 5 emisión, cable, etc. En algunos casos, el contenido creado por en una ubicación remota 13 puede ser transferido primero a un medio de almacenamiento, tal como un CD-RW, DVD, o dispositivo de memoria flash, y transportado a una emisora de radioemisión 16 a través de medios más convencionales dónde éste pueda ser almacenado en una base de datos u otro dispositivo de almacenamiento.

[0019] Después de su creación, el contenido de las fuentes 13, 14,15 y 60 se pueden entregar a los receptor(es) 30 a través de una red de transmisión de emisión. Esta red consiste esencialmente de una emisora de radioemisión 16 que ensambla el contenido de las fuentes 13, 14,15 y 60 y procesa el contenido como apropiado (p. ej., digitaliza, comprime, empaqueta), y una red de transmisión 17 que recibe el contenido 40 de emisora de radioemisión 16 y lo transporta 42 a dispositivo(s) receptores 30. En una forma de realización, la emisora de radioemisión 16 incluye software y/o hardware que se configura para procesar el contenido transmitido por fuentes 13, 14,15 y 60 como se ha descrito anteriormente. Un segundo mecanismo de entrega puede incluir una conexión 138 punto a punto directo entre receptor(es) 30 y fuente 18 que puede ser algún tipo de servidor. Esta conexión 138 puede ser hecha a través de una línea telefónica convencional, cable, inalámbrico, o de otra manera. Un tercer mecanismo de entrega también puede ser una conexión punto a punto 136, pero la transmisión del contenido de una fuente 19 para receptor(es) 30 se hace a través de una o más redes compartidas (p. ej., por medio de Internet).

[0020] Fig. 1 también ilustra que la emisora de radioemisión 16 puede ser opcionalmente acoplada a fuente 18 y/o fuente 19.

Tal acoplamiento puede permitir a la emisora de radioemisión 16 trabajar cooperativamente con la fuente 18 o fuente 19 en la transmisión del contenido a receptor(es) 30. También se ilustra en la Fig. 1 un backchannel (o canal de retorno) 26 por el que el receptor(es) 30 puede transportar a y/o recibir datos desde la emisora de radioemisión 16. El canal posterior 26 puede comprender una línea telefónica, cable, inalámbrica, u otra conexión.

[0021] Un mecanismo de entrega, la conexión directa de punto a punto a una fuente de contenido, puede comprender comunicación a través de una línea telefónica ordinaria. Este tipo de conexión típicamente es iniciada por el receptor(es) 30 para transmitir información a, o para recuperar información de un servidor de datos. Otro mecanismo de entrega, la conexión punto a punto a través de una o más redes, puede comprender una conexión típica entre nodos en Internet. Puesto que los datos se pueden dirigir a través de muchas redes compartidas diferentes en este caso, se puede leer, almacenar y escribir muchas veces ya que es transmitida desde la fuente 19 al receptor(es) 30. El tercer mecanismo de entrega puede incluir un satélite, cable o red terrestre de emisión 17. Se puede transmitir información desde y a receptor(es) 30 tanto en tiempo real como almacenar y enviar.

[0022] En una forma de realización, la emisora de radioemisión 16 además incluye un servidor proxy 21 que se configura para transcódicar el contenido recibido a un formato compatible con uno o más de dispositivos de cliente 30. Por ejemplo, proxy 21 puede recibir contenido basado en web incluyendo instrucciones escritas en el HTML, JavaScript™ (JavaScript es una marca registrada de Sun Microsystems, Inc), CSS, u otros idiomas, y transcribir el contenido recibido a un formato compatible con clientes 30. En una forma de realización alternativa, los clientes se pueden configurar para procesar directamente tales instrucciones. En tal caso, proxy 21 se puede configurar para ejecutar ciertos tipos de preprocesado del contenido antes de la transmisión a los clientes.

[0023] Pasando ahora a la Fig. 2, se muestra una visión de conjunto de una forma de realización de una emisora de radioemisión (cabecera) 16. La emisora de radioemisión 16 de la Fig. 2 incluye un servidor de aplicación 250 y una base de datos 230 que puede contener contenido interactivo previamente creado. También se muestra en la Fig. 2 una fuente 13 de contenido (p. ej., Internet) que es externa a la emisora de radioemisión 16 y acoplada a la emisora de radioemisión 16. La base de datos 230, el servidor 250, Internet 60, y la fuente 13 se acoplan a un mecanismo de procesado de contenido 200 que se configura para procesar el contenido recibido y transmitir el contenido procesado a un multiplexor 220. En la forma de realización ejemplar de la Fig. 2, el servidor proxy 21 incluye el servidor 250 y el mecanismo de tratamiento 200.

[0024] En una forma de realización el mecanismo de tratamiento de contenido 200 comprende un ordenador acoplado para recibir y transmitir contenido desde la fuente 13, la base de datos 230, o el servidor 250. El mecanismo de tratamiento 200 se configura para transmitir el contenido procesado al multiplexor 220. El multiplexor 220 también se acopla para recibir señales de audio/vídeo 240. El multiplexor 220 multiplexa las señales recibidas y transmite la señal multiplexada al operador de comunicaciones de red 17 dónde ésta es posteriormente transmitida a un dispositivo de recepción. Como se ha indicado anteriormente, proxy 21 se puede configurar para procesar contenido recibido antes de transmitir el contenido a los dispositivos de cliente. Por ejemplo, proxy 21 se puede configurar para recibir peticiones de clientes de contenido basado en web, obtener el contenido solicitado, y transcódicar el contenido recibido a un formato alternativo antes de la transmisión al cliente solicitante. Finalmente, además de lo dicho anteriormente, la emisora de radioemisión 16 incluye un procesador de datos de retorno 210 acoplado al backchannel 26. En una forma de realización, la procesador de datos de retorno 210 puede comprender un módem que recibe datos para procesamiento ulterior dentro de la emisora de radioemisión 16.

[0025] Mientras que la descripción anterior describe una fuente de contenido interactivo estando en una emisora de radioemisión 16, en una forma de realización alternativa la base de datos 230 y el mecanismo de procesado de contenido 200 puede residir en la ubicación de un operador de comunicaciones de red 17. Un ejemplo de tal forma de realización alternativa puede ser una estación de cable que inserta contenido interactivo en una señal de emisión antes de la transmisión. Tales numerosas alternativas son posibles y están contempladas.

[0026] Pasando ahora a la Fig. 3, se muestra una forma de realización de un dispositivo de recepción/inicio 1012, de ahora en adelante referido como un "cliente". Mientras que la Fig. 3 ilustra al cliente 1012 en forma de un descodificador de señales digitales 1012, el cliente 1012 también puede comprender otros dispositivos. En términos generales, el cliente 1012 se configura para recibir una primera señal 1070, tal como una señal de emisión, y transmite una segunda señal 1080, tal como a una pantalla o dispositivo de grabación. Mientras que en la forma de realización mostrada, el cliente 1012 se muestra acoplado a un dispositivo externo de almacenamiento en masa 1018, tal almacenamiento puede ser interno al mismo cliente 1012. El cliente 1012 incluye una unidad de control 1030, extremo frontal 1026, canal de retorno 1038, fase de transmisión 1028, y fase AV 1034. También se representa en la Fig. 3 una memoria 1080 que incluye OS y/o soporte personalizado 1044, un motor de procesamiento de mensaje 1036, y aplicaciones 1042. También se muestra una interfaz I/O 1040 y módulo(s) de acceso condicional (CA) 1032. La interfaz I/O 1040 se puede configurar para detectar la interacción del usuario a través de un control remoto, teclado, u otro dispositivo. La unidad de control 1030 puede comprender un microprocesador, memoria (por ejemplo; RAM), y otros componentes que son necesarios para desempeñar un cómputo multiuso ordinario.

[0027] En una forma de realización, las aplicaciones 1042, OS/soporte personalizado 1044, módulo(s) CA 1032, y motor de tratamiento de mensaje 1036 comprenden el código que se puede almacenar en un dispositivo de memoria del descodificador 1012. Adicionalmente, los módulo(s) CA 1032 pueden comprender software de sistema configurado para controlar el acceso a programas particulares o servicios que son accesibles por el descodificador 1012. Mientras que el motor de tratamiento de mensaje 1036 se muestra como código de programa que se puede almacenar en la memoria 1090 y ejecutado por la unidad de control 1030, se entiende que otras formas de realización son posibles y están contempladas. Por ejemplo, el motor de procesamiento de mensaje 1036 puede comprender circuitería o una combinación de hardware y software. Por ejemplo, el motor de procesado de mensaje 1036 puede comprender un dispositivo procesador ejecutando instrucciones de programa. Además, el motor de tratamiento de mensaje 1036 se puede configurar como un dispositivo externo que se puede acoplar a una unidad de recepción. Por ejemplo, tal dispositivo externo puede comprender un módulo de expansión que se configura para añadir funcionalidad de procesamiento de mensaje a un dispositivo preexistente.

[0028] En términos generales, el cliente 1012 es operable para recibir y descomprimir señales que pueden incluir datos digitales. Las señales descomprimidas se pueden convertir en señales analógicas tales como PAL, SECAM, o señales de formato NTSC para pantalla de televisión, o pueden estar en formato digital para su uso en una pantalla digital de televisión. Como se muestra en la Fig. 3, el cliente 1012 incluye circuitería de extremo frontal 1026 operable para recibir, audio, vídeo, y otros datos de una señal recibida 1070. La señal recibida 1070 se suministra al cliente 1012 en el extremo frontal 1026, que puede comprender un conversor de análogo a digital (A/D) y sintonizadores/desmoduladores (no mostrados). El extremo frontal 1026 puede seleccionar y pasar una frecuencia particular, desmodularla, y convertir señales analógicas a un formato digital. Mientras que los datos analógicos se pueden convertir en datos digitales, como se ha indicado anteriormente una señal recibida puede comprender datos digitales que pueden no requerir tal conversión. La salida digitalizada puede luego ser transmitida a una fase de transmisión 1028 que además procesa más los datos, transmitiendo una parte de los datos a una fase audio-visual (AV) 1034 para la pantalla y otra parte al procesador de control 1030. Además, el módulo CA 1032 puede recibir datos de la fase de transmisión 1028 y puede transmitir condicionalmente una señal descodificada u otra señal a la fase AV 1034. La señalización e información de control también se pueden incluir en la emisión junto con los datos audio-vídeo y se pueden manipular por software en el cliente 1012.

[0029] Las señales de audio-vídeo y las señales de control de programa recibidas por el cliente 1012 pueden incluir programas televisivos, metadatos, y selecciones de menú accesibles por un espectador a través de una interfaz de usuario, al igual que aplicaciones que pueden ser ejecutadas. Un espectador puede controlar al cliente 1012 de una variedad de maneras, incluido a través de una unidad de control remoto infrarrojo, un tablero de control en el cliente, o un dispositivo que se utiliza para elegir de un menú visualizado en la pantalla. Las selecciones y entradas hechas por el espectador se pueden destinar para una o más aplicaciones diferentes que se están ejecutando en el cliente. Como se ha mencionado anteriormente, las señales emisión 1070 se reciben a través del extremo frontal 1026 y se filtran por la fase de transmisión 1028. Las señales unicast o multicast se pueden recibir generalmente a través del canal de retorno 1038. Las aplicaciones 1042 que ejecutan en el cliente 1012 puede llegar allí mediante una variedad de maneras. Por ejemplo, las aplicaciones se pueden recibir a través de una señal emisión 1070, a través de la interfaz de recurso de canal de retorno 1038, o a través del dispositivo de almacenamiento 1018. Las aplicaciones recibidas a través del dispositivo de almacenamiento 1018 pueden haber sido enviadas originalmente con el cliente 1012 o pueden haber sido descargadas previamente desde otra fuente y almacenados en el almacenamiento 1018.

[0030] En una forma de realización, el cliente 1012 se puede configurar como un decodificador digital para su uso con un receptor de satélite o decodificador/receptor de satélite integrado que es capaz de descodificar vídeo MPEG, audio, y

datos.

Por ejemplo, el cliente 1012 se puede configurar para recibir canales de vídeo digital las cuales soportan comunicaciones de banda ancha usando Modulación de Amplitud en Cuadratura (QAM), Modulación de Fase en Cuadratura (QPSK), Multiplexación por División de Frecuencias Ortogonales Codificada (COFDM) o por Banda Lateral Vestigial 8 (VSB) y para controlar canales para mensajería y señalización en dos sentidos. Los canales digitales pueden llevar corrientes de transmisión MPEG de multiprogramas comprimidos y codificados (Motion Picture Expert Group). La fase de transmisión 1028 extrae el programa deseado de la corriente de transmisión y separa los componentes audio, vídeo y de datos, que se dirigen a dispositivos que procesan las corrientes, tal como uno o más decodificadores de audio, uno o más decodificadores de vídeo, y opcionalmente a RAM (u otra forma de memoria) o a un disco duro. Debe entenderse que el cliente 1012 y dispositivo de almacenamiento 1018 (así como cualquier dato y señales del proveedor de servicio de radiotransmisión) se pueden configurar para suministrar datos analógicos, digitales o tanto digitales como analógicos. Se puede realizar la conversión al formato digital para el almacenamiento de datos analógicos recibidos.

[0031] El dispositivo de almacenamiento 1018 se acopla opcionalmente al cliente 1012 y se puede configurar para almacenar vídeo, audio, código ejecutable, metadatos, y otros datos. El dispositivo de almacenamiento 1018 puede ser interno al cliente 1012 o conectado externamente (p. ej., a través de una conexión IEEE 1394-1995) bien con una conexión permanente o una conexión desmontable. Además, el dispositivo de almacenamiento 1018 puede comprender cualquier tipo adecuado de almacenamiento, tales como una unidad de disco duro, una unidad de DVD grabable, cinta magnética, disco óptico, disco magneto-óptico, memoria flash, o memoria de estado sólido. Además, se puede fijar más de un dispositivo de almacenamiento tal como el dispositivo 1018 al cliente 1012. El cliente 1012 y/o dispositivo de almacenamiento 1018 además puede ser incorporado en un aparato de televisión. Los datos ejecutables, tal como instrucciones del programa, que se almacena dentro del dispositivo de almacenamiento 1018 pueden ser recuperados y ejecutados. En una forma de realización, los datos recuperados se pueden ejecutar o utilizar de otra manera en sincronización con otras aplicaciones o señales recibidas, por ejemplo correspondiente a un programa de concurso, anuncio, o un juego de internet online. Alternativamente, los datos recuperados se puede ejecutar o ser utilizados independientemente, tal como para vídeo a demanda, actividades bancarias, correo electrónico, un navegador web, o una guía electrónica de programación (EPG).

[0032] Debe entenderse que el cliente 1012 y sistema 100 aquí descritos pretenden ser solo ejemplos. El sistema de emisión de red 100 y el cliente 1012 pueden ser diferentes a como se describen aquí sin apartarse del ámbito de la invención. Además, varios componentes representados en el cliente 1012 de la Fig. 3 pueden ser combinados, como la ubicación de la integración de dispositivo de almacenamiento 1018 dentro del cliente 1012. Numerosas alternativas son posibles y están contempladas.

#### 1. Modelo de aplicación y ciclo de vida

[0033] En términos generales, una aplicación televisiva interactiva puede iniciarse bien en un estado maximizado o estado minimizado, dependiendo de como esté creado. Señalar en el directorio puede indicar al sistema si la aplicación se está iniciando en estado minimizado o estado maximizado. Desde una perspectiva del sistema, la diferencia entre el estado minimizado y el estado maximizado es que las aplicaciones que están en el estado minimizado pueden no recibir una notificación cuando el espectador pulsa las teclas. Alternativamente, cuando en un estado maximizado, las aplicaciones pueden presentar un filtro al sistema que le dice al sistema que notifique cuando las teclas designadas en el filtro son pulsadas. Mientras no sea necesariamente requerido, una aplicación que se ejecuta en un estado minimizado típicamente reduce su uso de recursos. Por ejemplo, puede presentar un icono en la pantalla mejor que gráficos extendidos.

[0034] Además de lo dicho anteriormente, una aplicación en ejecución en bien un estado minimizado o estado maximizado puede ser suspendida. Tras la suspensión, generalmente no se notifica una aplicación, sino que ningún tiempo de cpu se asigna a la aplicación. Al finalizar la suspensión, una aplicación regresa al estado en el que estaba antes de la suspensión. En cualquier caso, el sistema puede invocar una función en la aplicación para notificar que la aplicación ha sido suspendida de modo que éste puede realizar cualquier acción necesaria para asegurar una consistencia interna.

[0035] Las aplicaciones puede terminar normalmente o el sistema puede pedir su terminación, por ejemplo, si una aplicación nueva aparece en la corriente de emisión. Debido a que una aplicación puede estar en un estado donde el término sería disruptivo para el espectador, la aplicación puede negar una solicitud para terminar inmediatamente. Por ejemplo, un espectador puede estar en medio de una transacción en línea para comprar un producto publicitado. Cuando una aplicación termina, el sistema es notificado de modo que éste pueda determinar, típicamente trabajando junto con el control de tarea proporcionado por la red, qué aplicación ejecutar a continuación.

[0036] Las transiciones entre estados pueden ser respuestas a una variedad de estímulos incluyendo señalización de emisión, pulsaciones de tecla del espectador, y decisiones hechas por el sistema o de las aplicaciones mismas. Como se ha indicado anteriormente, el estado inicial de una aplicación se puede determinar por señalización de emisión. Se puede usar un pulsado de tecla para causar que una aplicación se mueva del estado minimizado al estado maximizado. La aplicación misma puede decidir cuando pasar al estado terminado y cuando pasar al estado minimizado. El sistema puede suspender una aplicación para ejecutar otra aplicación. Además, la señalización de emisión puede provocar que

el sistema solicite que una aplicación salga.

[0037] Mientras que el ciclo de vida definido arriba puede representar un ciclo de vida por defecto, las modificaciones al ciclo de vida se pueden proporcionar mediante llamadas en un control de tarea de un proveedor de red. Por ejemplo, uno ni siquiera tendría necesidad de cargar una aplicación hasta que el espectador responda con una selección de botón apropiado en respuesta a la presentación de un icono. Además, el ciclo de vida anterior puede generalmente corresponder a un modelo en el que sólo una única aplicación es ejecutable a la vez. No obstante, para sostener aplicaciones múltiples al mismo tiempo, el modelo de aplicación y definición de ciclo de vida puede ser más complejo. Por ejemplo, las prioridades se pueden señalar de modo que la implementación pueda determinar qué aplicaciones se pueden ejecutar en el caso de que el hardware no sea capaz de soportar todas las aplicaciones señaladas simultáneamente.

[0038] Las aplicaciones desarrolladas para el uso en sistemas de televisión interactiva generalmente pueden incluir código de programación similar al de los lenguajes de programación tales como C, C++, etc. Sin embargo, con la proliferación de la World Wide Web (web), y el deseo de aprovechar la Web y los recursos de la Web en los sistemas de televisión interactiva, el uso de otros lenguajes tales como HTML y el lenguaje Javascript™ (JS) pueden ser útiles. No obstante, mientras que se puede desear el uso de aplicaciones de HTML, el ciclo de vida de las aplicaciones HTML en un entorno de televisión interactiva se puede complicar por factores diferentes.

[0039] Primero, el contenido de HTML/JS puede ser más dinámico que persistente. Por ejemplo, en un entorno de televisión interactiva actual, una aplicación se puede configurar para sólo ejecutar código o usar datos que se empaquetan en el carrusel en el mismo directorio que el primer módulo de programa. Por lo tanto, para cuestiones de seguridad el contenido del directorio puede definir claramente el límite de la aplicación y los permisos señalados en el directorio se pueden aplicar al contenido entero del directorio. No obstante, el contenido de HTML/JS puede referirse a otro contenido (p. ej., a través de un enlace) que debe ser adquirido de alguna ubicación diferente del carrusel y el contenido al que se refiere puede reemplazar el contenido inicial. No está claro que sea seguro en este caso aplicar los mismos permisos de seguridad para tal contenido de sustitución. Por lo tanto, debido a esta naturaleza dinámica, es más difícil definir un "límite de aplicación"

[0040] En segundo lugar, incluso cuando un producto no soporta múltiples aplicaciones concurrentes y restringe la aplicación a sólo aquel contenido soportado en el mismo directorio en el carrusel, puede haber expediciones de ciclos de vida que afectan la manera en que el autor de contenido diseña el contenido HTML/JS. Por ejemplo, si se determina que el emisor puede señalar que una aplicación pueda ser terminada, puede ser útil pedir un encargado por escrito por el autor de contenido para responder a tal evento. De forma similar, puede haber otros estados que se manejarían de forma óptima mediante un encargado específico de aplicaciones. Por ejemplo, si el espectador está en medio de una transacción que implica una aplicación, esta aplicación puede querer retrasar su término hasta que la transacción esté completa. Por lo tanto, una aplicación puede ser notificada por el sistema cuando la emisión señala una aplicación nueva disponible en la emisión. En una forma de realización, la aplicación puede ser notificada a través de un evento, tal como el evento `O_exit` identificado más abajo. Una aplicación que determina que no quiere terminar inmediatamente puede extender su vida mediante una llamada a una función de evento definido tal como "preventDefault" (evitar aplicación por defecto) ().

```
O_exist
Burbujas : si -- (véase modelo de eventos DOM)
Cancelable : si
Info de contexto: la razón por la que termina.
```

## 2. Sintonización y selección de secuencia

[0041] En una forma de realización se proveen dos formas diferentes para la sintonización de la señal de emisión y selección de secuencia. La primera usa un lenguaje de marcado, tal como HTML, y presupone que el autor de contenido tiene suficiente conocimiento a priori como se describe abajo. El segundo usa un lenguaje de escritura tal como JavaScript, no presume el mismo conocimiento a priori, y es suficientemente genérico como para ser aplicable a la selección de secuencia desde un disco duro local o VOD. Ambos hacen uso de un URL nuevo, definido aquí como la "emisión:" URL. Primero se describe el URL que se usa en ambos métodos .

El URL que se puede usar para sintonizar y seleccionar secuencia.

[0042] En algunos ambientes de emisión, tal como un ambiente basado en MPEG, puede ser posible asociar un único identificador global (o al menos red) a una secuencia de emisión. El uso de tal identificador único dentro de un esquema URL puede permitir la identificación única de recursos dentro de ese flujo. Una sintaxis de un esquema de emisión URL está provista más abajo. En términos generales, este esquema puede proporcionar un mecanismo general para identificar recursos de emisión de una manera que sea independiente de red y de plataforma. Este esquema puede trabajar con programas grabados digitalmente así como con emisiones en directo.

[0043] Lo siguiente es una sintaxis formal, en BNF como gramática, para una "emisión:"URL. En lo que sigue, nótese

que las reglas están separadas de las definiciones por un igual "=", "|" se utiliza para designar alternativas, los literales se citan con "", los paréntesis "(" and ")" se utilizan para agrupar elementos, los elementos opcionales se incluyen en corchetes "[" y "]", y los elementos pueden ser precedidos por "<n>\*" para designar n o varias repeticiones del siguiente elemento donde n tiene el valor predeterminado 0.

5

```

broadcast_url_ = broadcast_scheme ":" [broadcast_hier_part]
broadcast_scheme_ = "broadcast"
broadcast_hier_part_ = broadcast_net_path | broadcast_abs_path
broadcast_net_path_ = "/" service_address [component_list ] [ broadcast_abs_path ]
service_address_ = channel_name | "current"
channel_name_ = *( domainlabel ".") toplabel
domainlabel_ = alphanum | alphanum *( alphanum | "-" ) alphanum
toplabel_ = alpha | alpha *( alphanum | "-" ) alphanum
alphanum (may be as defined in RFC 2396)
alpha (may be as defined in RFC 2396)
component_list_ = ";" component *( "," component)
component_ = stream_selector
stream_selector_ = stream_type "=" stream_id
stream_type_ = "video" | "audio," | "data" | "subtitle" | "teletext"
stream_id_ = 1 *alphanum | "default" | "current" | "none"
broadcast_abs_path_ = "/" path_segments
path_segments_(may be as defined in RFC 2396)
    
```

[0044] Dada la definición anterior, puede ser representado un ejemplo de un resumen de uso :

10 **broadcast: { // <service address>; <component list> }**

[0045] Donde *service\_address* es definido de la siguiente manera:

*service\_address* ::= *channel\_name* *current*

donde:

15 *channel\_name* especifica un nombre estilo DNS que identifica únicamente el canal.

*current* especifica el servicio actualmente seleccionado.

La *component\_list* es una lista separada por comas que identifica componentes específicos en la secuencia y puede ser definida como sigue:

20 *component\_list* ::= *component* \*( "," *component* )

*component* ::= *stream\_type* "=" ( *track\_tag* | "default" ) |

*stream\_type* ::= "video" | "audio" | ...

Un *track\_tag* se puede definir como un string ASCII de longitud arbitraria, típicamente entre 1 y 4 bytes. Un *track\_tag* de "0" es equivalente al componente de valor predeterminado del tipo de secuencia especificada. Por ejemplo, el URL "broadcast://tfl.fr; video=0/" broadcast://tfl.fr; video=0, audio=eng" identifica la secuencia de vídeo de valor predeterminado y la secuencia de audio inglés en el canal nombrado "tfl.fr".

25

Tipos A/V MIME asociados a la emisión: Url

30 [0046] Mientras que la siguiente exposición describe principalmente la semántica asociada a tipos de secuencias de vídeo y de audio, se permiten otros tipos de secuencias dentro de URL y se tratan en la sección titulada "Obteniendo Aplicaciones y Datos" más abajo. En cualquier caso, en una forma de realización, los siguientes eventos se pueden despachar durante una selección de servicio.

**Eventos de selección de secuencia.**

35

[0047]

- 40 **Click** La selección ha ocurrido como resultado de un evento de clic. La acción por defecto es solicitar el servicio especificado. Notar que este evento generalmente es un evento de entrada de usuario.
- Load** Si la solicitud sucede, un evento de carga será despachado. Normalmente tal evento es despachado después de que el URL se haya terminado de cargar, pero las secuencias de audio y vídeo indefinidos nunca se acabarían de cargar. Por lo tanto, es apropiado despachar este evento una vez que se haya sido iniciado con éxito el procesamiento de las secuencias de audio y vídeo solicitadas a través de todo el hardware implicado en la tubería procesadora.
- 45 **Error** Si la solicitud es denegada o de otra manera inválida se despacha un evento de error.
- Abort** Si el usuario aborta la solicitud antes de despachar el evento de carga, se despacha un evento de abortar.
- Unload** Si la solicitud reemplaza un objetivo existente, se despacha un evento de descarga.

[0048] Cuando no se especifica ninguna lista de componentes, el tipo MIME correspondiente a la emisión:URL puede ser `application/mpeg.service` y este tipo puede corresponder a un servicio tal y como se define como MPEG estándar. Por lo tanto, tal tipo MIME contendría no sólo vídeo, audio, y subtítulos, sino también los datos que se multiplexan en el mismo servicio con éstos, por ejemplo, HTML y/o otras aplicaciones.

[0049] Cuando se especifica un componente de vídeo, el tipo MIME correspondiente a la emisión:URL es `video/mpeg`. De forma similar, cuando se especifica un componente de audio, el tipo MIME correspondiente a la emisión:URL es `audio/mpeg`.

[0050] Como se muestra en los ejemplos más abajo, es posible referirse a múltiples secuencias elementales en un URL único. Si las secuencias indicaron incluir sólo un único flujo de vídeo y un único flujo de audio que se sincroniza con aquel flujo de vídeo, entonces las secuencias resultantes serán considerados del tipo `video/mpeg`; o si no, el tipo de los flujos múltiples será del tipo `application/mpeg.service`.

## EJEMPLOS

[0051] Esta sección explica el significado de diferentes ejemplos de URLs, que en algunos casos, si usados como el ejemplo completo mostrado en la siguiente sección, podrían resultar en una selección de sintonización o secuencia.

`broadcast:`

Identifica el `service_address` actualmente sintonizado y el `component_list` para la tubería primaria (ver Sintonización JS y Selección de Flujo más abajo). Este uso es similar al "tv:" en la especificación DASE y "dvb://current.av" en la especificación MHP. Así, por ejemplo, este se puede utilizar dentro de un elemento de HTML para redimensionar y reubicar el vídeo que se está reproduciendo en ese momento.

`broadcast://cnn.com`

Identifica el canal de TV de CNN y todas sus secuencias de componentes. Esta forma de URL puede utilizarse para solicitar que el sintonizador de la televisión cambie de canal. Este URL en un contexto de selección de servicio causa la selección automática de las secuencias de valor preestablecido. Esto es, cuando se usa en un contexto de selección de servicio, el agente-usuario (si la aplicación está apropiadamente autorizada) sintonizará el canal nuevo y automáticamente seleccionará la secuencia de vídeo predeterminada, seleccionará la secuencia de audio predeterminada (basado en el lenguaje preferido), seleccionará los subtítulos preestablecidos y el teletexto si está identificado en las preferencias de stream del usuario, y seleccionará el carrusel de datos preestablecido.

`broadcast://cnn.com;audio=eng`

Identifica el canal de tv de CNN y selecciona explícitamente sólo la secuencia de audio inglesa. Los documentos usan esta forma de URL para referirse explícitamente a una secuencia elementalmente específica.

`broadcast://current;audio=eng,Video=current`

Selecciona la secuencia de audio inglesa en el servicio actual. Este URL permite al autor cambiar la secuencia de audio actual sin saber explícitamente la dirección del servicio actual, y sin cambiar la secuencia de vídeo actual.

### 2.1 Sintonización de HTML y selección de secuencia.

[0052] Cuando el autor de contenido tiene conocimiento del nombre DNS que corresponde a un canal dado, pueden usar HTML para ocasionar una sintonización a ese canal. Por ejemplo, la siguiente HTML permite que el actual documento de HTML presente un enlace, "my\_link", que solicita al sintonizador que seleccione un nuevo canal.

```
<A ID="my_link" HREF="broadcast://cnn.com">Click Me</A>
```

[0053] Si la solicitud es autorizada y se resuelve a un nombre canal válido luego el documento de HTML será descargado y sustituido con un operario de medios televisivos que reproduce el vídeo predeterminado y los flujos audio asociados al `cnn.com` service. En una forma de realización, las aplicaciones de HTML pueden permitir el uso de URLs que hacen referencia a vídeo de MPEG o flujos de audio o servicios de MPEG- 2 como se ilustra en los siguientes elementos de HTML y atributos CSS. Si el uso del URLs resulta en una selección de componentes del servicio actualmente sintonizado, solo pueden ocurrir los eventos de Carga, Error, o Aborto.

Atributo	Elemento HTML		
	video/mpeg	audio/mpeg	application/mpeg.service
Imagen del fondo	si		si
Vídeo del fondo	si		si
a.href	si	si	si
img.src	si	si	si

input.src	si	si	si
object.data	si	si	si

5 Además, los URLs pueden causar selección de servicio cuando son referidos a través del objeto de ubicación en un modelo de objeto de documento de escritura como se describe abajo o cuando se usa como un parámetro en un diálogo "goto".

## 2.2 Sintonización de JavaScript y Selección de secuencia

10 [0054] Una segunda manera de habilitar la sintonización de señal y la selección de flujo utiliza un lenguaje de escritura tal como JavaScript para permitir al programador de contenido controlar explícitamente tuberías virtuales que existen entre fuentes de audio y vídeo (p. ej., sintonizador, disco duro local) y sus destinos (p. ej., pantalla, disco duro local). Esta sección describe como un programador de JavaScript puede ejercer un control de alto grado sobre no sólo qué flujos se eligen para mostrar, sino también sobre qué flujos se pueden grabar sobre un disco duro y la velocidad y dirección con las que los flujos grabados son visualizados.

15 [0055] Una abstracción, conocida como una tubería, se puede utilizar para representar la asociación entre la fuente de una secuencia (p. ej., un sintonizador o un fichero que contiene un registro en un disco duro) y el destino definitivo (p. ej., la pantalla o un fichero en el disco duro), incluyendo, por ejemplo, cualesquiera recursos que estén requeridos entre la fuente y destinación (p. ej., hardware de acceso condicional, búferes I/O).

20 [0056] Cuando el software del receptor se inicia, se pueden definir un conjunto (o variedad) de tuberías. En una forma de realización, este conjunto de tuberías representa todas las conexiones posibles entre las fuentes de flujo y los destinos que se pueden representar en una plataforma de hardware en particular. Otras formas de realización pueden representar menos que todas las conexiones posibles. A causa de estas abstracciones, es posible tener una tubería definida sin tener todo el hardware que habitualmente requiere la tubería actualmente asignada a esa tubería en particular. Una tubería definida donde se ha ubicado menos que todo el hardware se dice que está en un estado "no realizada". Una tubería está "realizada" cuando todo el hardware requerido ha sido asignado a esa tubería. El programador puede usar el conjunto de tuberías definidas para:

- 30
- seleccionar una tubería
  - ajustar la fuente de una tubería
  - ajustar la destinación de una tubería si es un fichero
  - controlar la velocidad de una tubería, si la fuente es por lo tanto controlable, y también es capaz de ajustar la ubicación cuando sea posible

35

  - seleccionar los componentes de un flujo que será enviado a la destinación
  - añadir o eliminar escuchas de evento
  - y solicitar que se comience una tubería nueva para usos de grabación.

40 Además, el programador puede determinar que tubería es usada para una imagen dada usando la identidad que se une con aquella imagen. Por ejemplo, si hay un recorte de HTML incluido que declara

```

```

45 después, el programador de JS puede referirse a foo.pipe e invocar cualquiera de los método descritos abajo y puede leer/escribir los valores en los atributos como permitidos por la definición de abajo.

### La colección de tuberías y el objeto Tvpipes

50 [0057] Referencia de modelo de objeto:

```
[window].navigator.tv.pipes[i]  
[window].navigator.tv.pipes.primary
```

55 [0058] El conjunto de tuberías arriba es una colección de objetos Tvpipes como se describe abajo. El objeto primario es una referencia a un objeto de tubería que puede ser ajustable o recogible en JavaScript. El objeto Tvpipes tiene las siguientes propiedades, métodos, y colecciones.

### El objeto TVPipe

#### 60 Propiedades:

[0059] name String que identifica esta tubería en la gama de tuberías. (solo lectura)

src URL correspondiente al canal actual (lectura/escritura) --puede corresponderse bien con un archivo: o emisión: url

realizado "verdadero" | "falso" (solo lectura)

5 estado "conectado" | "en conexión" | "desconectado" "en desconexión" -- (solo lectura)

destino solo si la tubería está siendo usada para grabar (lectura/escritura) -- url correspondiente a archivo:

10 tipo "grabar" | "visualizar" (solo-lectura)

posición sin firmar int (lectura/escritura)  
-- # de ms en el evento

15 velocidad int (lectura-escritura)  
-- 100 es velocidad normal  
-- 0 está parado  
-- 500 es 5 veces la velocidad normal  
-- -100 es la velocidad normal, hacia atrás

20 -- -500 es 5 veces la velocidad normal, hacia atrás  
-- 50 es la mitad de la velocidad, hacia adelante, etc.

event\_info parejas del valor del nombre sobre el evento actual (solo lectura)

25 **Colecciones:**

[0060] Componentes[] gama de objetos de componentes (ver **TvComponent** debajo) indicando aquellos que son seleccionados habitualmente

30 **Métodos:**

[0061] record(uri) inicia la grabación al archivo nombrado en el uri si existen los recursos suficientes.  
addEventListener()  
removeEventListener()  
35 dispatchEvent()

**Objeto TvComponent**

40 [0062] Un objeto TvComponent representa una secuencia de datos que puede llevar vídeo, audio, datos interactivos, subtítulos, u otro tipos de contenido.

[0063] Referencia de modelo de objeto:

45 [window].navigator.tv.pipes[i].components[i]

[0064] Propiedades:  
name String que representa el nombre (es decir, el valor del track\_tag) del componente (solo lectura)

50 "true" seleccionado | "false" (lectura/escritura) -- booleano que indica que este componente ha sido seleccionado

type "audio"|"video"|"data"|"subtitles"|"teletext" ... (solo lectura)

**3. Control de la pantalla y reproducción de vídeo, gráfico y audio en HTML/JS**

55 [0065] Esta sección describe como se pueden situar y dimensionar gráficos encima del vídeo, como el vídeo puede ser situado y dimensionado por si mismo, y como puede ser controlado el audio. También se discute aquí la transparencia entre el plano de gráficos y el plano de vídeo, color basado en paleta, y el I-Frame de MPEG.

60 [0066] En una forma de realización, un receptor se puede configurar para soportar múltiples gráficos y capas de vídeo. En tal forma de realización, puede haber un capa en el nivel más bajo que se utiliza para mostrar vídeo y una capa interactiva (OSD) encima que se utiliza para mostrar texto y gráficos. La renderización del vídeo, tanto instantáneas (por ejemplo; I-Frames) como vídeo en movimiento, pueden ser soportados por un decodificador de hardware de MPEG.

65 [0067] Además de lo dicho anteriormente, una extensión puede sostener una capa encima de la capa OSD llamada la capa de subtítulo. Otra extensión puede ser usada para sostener una capa de gráficos multi-plano. En una forma de

realización, esta capa puede encontrarse lógicamente entre la capa del nivel más bajo y la capa interactiva. Esta capa de gráficos multi-plano se puede utilizar para mostrar dibujos estáticos tales como JPEG, MPEG u otras imágenes. Abajo se incluye una discusión de soporte para imágenes en la capa de gráficos multi-plano.

### 5 3.1 Color y transparencia

[0068] Existen varios modelos para especificar como se representa la información de color. Por ejemplo, un "espacio de color" es un modelo para representar color en cuanto a valores de intensidad. Ejemplos de espacio de color incluyen RGB, comúnmente usado para monitores informáticos, CMYK que se usa para impresoras de color, y YUV que es generalmente usado para televisión.

[0069] El número de bits usado para definir el color de un píxel puede ser referido como su profundidad de bit. El color real, a veces referido como color de 24 bit, es la especificación del color de un píxel en una pantalla que usa un valor de 24 bit. Usando 24 bit para especificar color, hasta 16,777,216 colores son posibles. Los sistemas de visualización varían en su capacidad para soportar color. Por ejemplo, algunos sistemas de visualización de color ofrecen un modo de color de 32 bit. En un sistema de pantalla de color de 32 bit, el byte extra, llamado el canal alfa, se puede utilizar para control e información de efectos especiales.

[0070] Porque los descodificadores de gama baja pueden no tener memoria suficiente para soportar color real, se pueden usar modelos basados en paleta. Con un modelo en base de paleta, el color de un píxel se representa por un índice en una paleta de colores. En tal modelo, los autores de contenido pueden definir paletas de color propios que contiene colores de su elección. Los colores reales en una paleta típicamente se representan como números de 48 bit con los primeros tres de aquellos números representando el color real y el cuarto de los números representando la cantidad de transparencia en el color.

[0071] En un sistema donde hay memoria suficiente para soportar color real, diversas aplicaciones pueden compartir la pantalla sin apenas problemas debido a que la paleta de colores fija es suficientemente grande para ajustar las diferentes tonalidades múltiples requeridas por cada aplicación. No obstante, en un sistema donde el número de colores soportable está limitado, si múltiples aplicaciones que comparten la pantalla establecen su propia paleta de colores, la sensación del espectador puede ser molesta.

[0072] Frecuentemente los dispositivos donde los gráficos solapan el vídeo (tales como descodificadores más baratos) tienen paletas con modelos limitados de transparencia incorporada. Dos modelos comunes donde la transparencia está limitada incluyen lo siguiente:

a. Sólo se soporta un único elemento no opaco en la paleta. Por ejemplo, este elemento podría ser completamente transparente, o podría ser un rosa transparente al 50%, etc. En cualquier caso, todos los otros elementos deben ser opacos.

b. Se soporta un único elemento en la paleta que puede ser semitransparente o completamente transparente. Todos los demás elementos de la paleta pueden ser o bien completamente opacos o poseer una cantidad particular de transparencia fija. Por ejemplo, una paleta que puede poseer  $n$  colores podría presentar un único color que es transparente al 30%,  $m$  ( $m > 1$ ) colores que son transparentes al 50%, en este caso el resto de colores  $n-(m+1)$  debe ser bien transparente al 50% o bien completamente opaco. En otras palabras, no puede haber 3 colores no opacos en una paleta donde todos poseen un nivel diferente de transparencia.

[0073] Para maximizar la disponibilidad de los valores de transparencia para el uso del autor, se puede definir un sistema que permita a un autor especificar una región, incluyendo tanto su ubicación como dimensiones, que quieran para contener gráficos solapados. Si el autor no fuera capaz de especificar esta región, se tendría que "retirar" (el) único color transparente pintando el área exterior de la región de gráficos con el (a veces único) color transparente disponible en la paleta. (Esto también reduce la cantidad de espacio requerido para almacenar los gráficos OSD). Posteriormente, la aplicación se puede configurar para cambiar dinámicamente su región (incluso cuando la aplicación es transcodificada antes de la emisión).

#### 55 Paleta variable-fija

[0074] En una forma de realización, se puede utilizar una paleta de combinación variable-fija donde los componentes variables son especificados por la aplicación. El primer color  $m$  de  $n$  se puede elegir para ser fijado en el color 0 siendo completamente transparente. Por ejemplo, en una paleta de 256 colores donde hay 8 bits disponibles para color, los primeros 188 colores pueden encontrarse, como se ha especificado, en un estándar propuesto o existente, como la paleta de colores de DVB MHP. Los restantes 68 colores pueden tomarse de colores especificados por la paleta de colores acompañando a la imagen. En una forma de realización, estos 68 colores se pueden seleccionar desde los primeros 68 colores específicos en la paleta de imagen. Por lo tanto, un diseñador de contenido de aplicación debería asegurarse que los colores más importantes están colocados los primeros en la paleta.

[0075] Si es necesario soportar múltiples aplicaciones, cada una de las cuales posee su propia paleta de colores,

entonces el sistema puede elegir colocar en la paleta una mezcla de los primeros colores en cada una de las paletas específicas de la aplicación/imagen. De forma similar, en cualquier momento se espera que múltiples imágenes compartan la pantalla, el autor de aquellas aplicaciones puede obtener mejores resultados usando sólo los colores fijos en una de las imágenes o la misma paleta para ambas imágenes.

[0076] La transparencia entre los gráficos y los planos de vídeo puede ser importante en la televisión interactiva, ya que normalmente el espectador quiere ser capaz de ver el vídeo que está reproduciéndose mediante el texto interactivo o las imágenes. En una forma de realización, las reglas de composición SRC Porter-Duff se pueden utilizar para componer gráficos entre sí. Generalmente, el vídeo subyacente es opaco, por lo tanto el vídeo se muestra a través de los gráficos cuando estos son transparentes. La regla de composición Porter-Duff SRC es relativamente fácil de computar debido a que la transparencia de un objeto sobre otro elige el valor alfa del objeto (transparencia) en la parte superior debido a la transparencia de los objetos compuestos. Mientras que en algunos casos este resultado puede parecer algo antinatural, los artistas gráficos están acostumbrados a planificar su diseño gráfico con esta regla en mente.

[0077] Como puede ser computacionalmente complejo el hecho de computar el valor alfa resultante, se puede permitir a los descodificadores aproximarse a la regla SRC-Over usando la regla SRC (a menos que el objeto de la parte superior sea completamente transparente, en cuyo caso los valores de píxel para el objeto transparente no deberían ser aplicados). En una forma de realización, las aplicaciones de HTML pueden especificar una regla de composición predeterminada, tal como SRC-Over. Sin embargo, en aquellos casos en los que un descodificador no tiene potencia computacional suficiente para computar la composición SRC-Over, se puede usar una aproximación a la regla SRC-Over (por ejemplo, usando la regla SRC Porter-Duff).

### 3.1.1 La propiedad clut

[0078] El formato de paleta analizado abajo permite imágenes cuyos colores se especifican usando un índice en una paleta para especificar también valores de transparencia por píxel mediante el uso de un canal alfa. No obstante, para otras imágenes, bases, etc., se puede requerir otro método para especificar la transparencia. Por lo tanto, las nuevas propiedades que permiten la especificación de estos valores alfa se describe en la subsección llamada a continuación "Propiedades Alfa".

[0079] Un autor de una aplicación puede especificar que una paleta particular (frecuentemente referida como una tabla de consulta de color o abreviado "clut") puede ser útil en la renderización de objetos en el cuerpo de una página HTML. Esta paleta podría ser usada de una manera entre varias diferentes. Por ejemplo, en una red vertical el autor puede especificar tanto una paleta como los colores de objetos usando sólo esa paleta porque éstas saben que todos los receptores tienen capacidades de color similares. Alternativamente, cuando el autor espera que su aplicación pueda ser usada en una red que incluye receptores de diferentes capacidades, esta paleta puede servir como indicio en cuanto a los mejores colores a usar. En cualquier caso el autor puede especificar una paleta de colores usando la propiedad "clut" documentada a continuación.

	"clut"
Valor	: <url>  ninguno
Inicial	: valor predeterminado seleccionado
Se aplica a	: cuerpo
Heredado	: si
Porcentaje	: N/A
Tipo de medio	: tv

El valor <url> de arriba se puede utilizar para identificar la ubicación de la paleta actual. Si no se especifica un valor <url>, o no hay propiedad "clut" en la hoja de estilo o en línea, se puede usar una paleta predeterminada.

[0080] En la tabla que sigue se presenta una forma de realización de un formato de paleta. En una forma de realización, el tipo MIME asociado a un url que contiene una paleta en el formato definido por la tabla siguiente puede ser "application/clut," con una extensión de ".clt". Además, los agentes de usuario y las aplicaciones de HTML pueden aceptar cluts en el formato usado para imágenes "png". Los tipos de estos cluts puede ser iguales que todas las imágenes png.

Ejemplo de uso (usando estilo en línea):

```
<BODY style="http://cnn.com/demoClut.clt">
```

**Formato de paletas de tipo aplicación/clut:**

[0081]

	N.º de bits	Identificador	Notas
PaletteResource() {			
color_model	8	uimsbf	El valor de 1 para el modelo de color puede usarse para indicar RGB, mientras que el valor 2 se usa para indicar YUV.
nub_colors	16	uimsbf	El valor en nb_colors es el número de colores de la paleta.
first_color	8	uimsbf	El propósito del valor de first_color es permitir múltiples recursos, cada uno especificando su propia paleta, para compartir el espacio de color.
for (i=0; i<n; i++){			La primera, segunda y terceras cantidades (amt_first, etc.) se refieren a la cantidad de RGB o YUV, dependiendo del valor de color_model. El valor en alfa (amt_transparency) representa la transparencia con 0 indicando transparencia y 225 indicando opacidad.
amt_first	8	uimsbf	
amt_second	8	uimsbf	
amt_third	8	uimsbf	
amt_transparency	8	uimsbf	
}			
}			

5

**3.1.2 Propiedades alfa**

[0082] El uso de una paleta específica de aplicación permite a un autor especificar el canal alfa que corresponde a un índice particular. A continuación hay una forma de realización que ilustra cómo las propiedades alfa pueden ser especificadas.

10

“alpha”

Valor : <hexadecimal-integer> | <percentage> | <normalized-number>  
 Inicial : #FF  
 Se aplica a : All elements  
 Heredado : yes  
 Porcentaje : percent opacity  
 Tipo de medio : tv

[0083] Ejemplo de uso:

15 <EM color=#008080 style="alpha:#C0">

En una forma de realización, el valor #FF es completamente opaco y el valor #00 es completamente transparente. El número normalizado puede variar entre 0.0 (completamente transparente) y 1.0 (completamente opaco). De forma similar, el 0% puede indicar transparencia completa y el 100% opacidad completa. Estos mismos términos se pueden utilizar con significados similares en las propiedades adicionales ilustradas a continuación.

20

“background-alpha”

Valor : <hexadecimal-integer> | <percentage> | <normalized-number>  
 Inicial : #FF  
 Se aplica a : All elements  
 Heredado : no  
 Porcentaje : percent opacity  
 Tipo de medio : tv

[0084] Ejemplo de uso:

25 <BODY style="background: black; background-alpha:#00">

“border-alpha”

Valor : <hexadecimal-integer> | <percentage> | <normalized-number>  
 Inicial : #FF  
 Se aplica a : All elements  
 Heredado : no  
 Porcentaje : percent opacity  
 Tipo de medio : tv

"border-top-alpha"

Valor	: <hexadecimal-integer>   <percentage>   <normalized-number>
Inicial	: #FF
Se aplica a	: All elements
Heredado	: no
Porcentaje	: percent opacity
Tipo de medio	: tv

"border-bottom-alpha"

Valor	: <hexadecimal-integer>   <percentage>   <normalized-number>
Inicial	: #FF
Se aplica a	: All elements
Heredado	: no
Porcentaje	: percent opacity
Tipo de medio	: tv

"border-left-alpha"

Valor	: <hexadecimal-integer>   <percentage>   <normalized-number>
Inicial	: #FF
Se aplica a	: All elements
Heredado	: no
Porcentaje	: percent opacity
Tipo de medio	: tv

"border-right-alpha"

Valor	: <hexadecimal-integer>   <percentage>   <normalized-number>
Inicial	: #FF
Se aplica a	: All elements
Heredado	: no
Porcentaje	: percent opacity
Tipo de medio	: tv

"outline-alpha"

Valor	: <hexadecimal-integer>   <percentage>   <normalized-number>
Inicial	: #FF
Se aplica a	: All elements
Heredado	: no
Porcentaje	: percent opacity
Tipo de medio	: tv

5

### 3.2 Posición de gráficos sobre el vídeo

[0085] Un programador de HTML puede usar hojas de estilo de cascada (CSS) para especificar la posición relativa o absoluta de los gráficos sobre el vídeo. Adicionalmente, el CSS se puede utilizar para especificar otras características también, tales como un margen, asociado a la apariencia visual de un gráfico o bloque de texto.

[0086] En una forma de realización, el tamaño del OSD se puede definir como el tamaño del bloque (div) cuyo nombre se ha definido como "osd". Si no se presentan dichos bloques, el tamaño puede ser el tamaño de la primera división en una ventana de nivel superior. Donde un descodificador no puede crear un OSD de ese tamaño exactamente, se puede utilizar el tamaño disponible más cercano al tamaño especificado. Los ejemplos siguientes ilustran cómo los gráficos se pueden situar en relación al vídeo de fondo. La pantalla resultante para cada uno de los ejemplos es la misma, dadas las presunciones constatadas a continuación en las descripciones.

[0087] En este primer ejemplo, el fondo se fija a un vídeo emitido a través de un url usando un atributo de imagen de fondo. En este caso se presupone que a la aplicación se le han concedido los privilegios de sintonización y, por lo tanto, el sintonizador se sintoniza en la estación que lleva la red Family-Videos y se visualizan el vídeo y el audio predeterminados.

#### 25 Primer ejemplo de posición de imágenes sobre el vídeo

[0088]

```

30 <html>
    <head>
        <title>example</title>

```

```

</head>
<body style="background-image: url(broadcast://family-videos.com)">
<div style="position: absolute; left: 200px; top: 80px; color=gray; border: thin solid
red">
5 <div style="position: absolute; left: 10px; top: 10px; font-size=18pt;
    line-height=120%; color=yellow; border: thin solid yellow;
    background-alpha: #01; compose-rule: src">
    <p>Nicolas a 18 mois
10 </div> 
</div>
</body>
</html>

```

[0089] En el segundo ejemplo, se presupone que la televisión ya ha sido sintonizada en la red Family-Videos.

#### Segundo ejemplo de posición de imágenes sobre vídeo

[0090]

```

20 <html>
    <head>
    <title>example</title>
    </head>
    <body style="background-image: url(broadcast://current); ">
25 <div style="position: absolute; left: 200px; top: 80px; color=gray; border: thin solid
    red">
    <div style="position: absolute; left: 10px; top: 10px; font-size=18pt; line-
        height=120%; color=yellow; border: thin solid yellow;
        background-alpha: #01; compose-rule: src">
30 <p>Nicolas a 18 mois
    </div>
    
    </div>
    </body>
35 </html>

```

[0091] En el tercer ejemplo, se vuelve a presuponer que la televisión ya ha sido sintonizada en la red Family-Videos y se ha seleccionado explícitamente un color transparente para los fondos (aunque de todos modos esto sería lo predeterminado).

#### Tercer ejemplo de posición de imágenes sobre vídeo

[0092]

```

45 <html>
    <head>
    <title>example</title>
    </head> <body style=" background-color: transparent">
    <div style="position: absolute; left: 200px; top: 80px; color=gray; border: thin
50     solid red">
    <div style="position: absolute; left: 10px; top: 10px; font-size=18pt;
        line-height=120%; color=yellow; border: thin solid yellow;
        background-alpha: #01; compose-rule: src">
    <p>Nicolas a 18 mois </div>
55 
    </div>
    </body>
    </html>

```

[0093] El cuarto ejemplo muestra que los antecedentes no requieren ser especificados en absoluto, suponiendo de nuevo que la televisión ya haya sido sintonizada en la red Family-Videos.

#### Cuarto ejemplo de posición de imágenes sobre vídeo

[0094]

```

5 <html>
  <head>
    <title>example</title>
  </head>
  <div style="position: absolute; left: 200px; top: 80px; color=gray; border: thin
    solid red"> <div style="position: absolute; left: 10px; top: 10px; font-
    size=18pt;
      line-height=120%; color=yellow; border: thin solid yellow;
      background-alpha: #01; compose- rule: src">
10 <p>Nicolas a 18 mois
  </div>
  
  </div>
  </body>
15 </html>

```

A algunos descodificadores les pueden faltar los recursos para reproducir videos y mostrar un OSD completo al mismo tiempo. Por lo tanto, para responder ante esta posibilidad, una aplicación de HTML en uno de estos descodificadores puede intentar no interpretar contenido en estos descodificadores a menos que un META elemento, como se muestra a continuación, se utilizada para indicar que el contenido fue diseñado específicamente para estos descodificadores. Meta-datos de cabecera:

```
<META name="tv-use" content="full-screen">
```

### 25 3.3 Cuándo renderizar gráficos

[0095] Cuando se renderizan los gráficos mientras se descargan, a veces tiene sentido retrasar la visualización al espectador hasta que al menos un subconjunto de los recursos, que han sido estimados como esenciales por los creadores de contenido, hayan sido descargados. En una forma de realización, un creador de contenido puede etiquetar el subconjunto esencial de recursos identificándolos usando una instrucción tal como una cabecera de meta-datos "requerida previamente". Por ejemplo, lo siguiente indica que la renderización de la página no puede ocurrir antes de llegar a "background.mpg"

```
<META name="prerequisite" content="http://www.enn.com/background.mpg">
```

[0096] Además de indicar que ciertos recursos pueden ser requeridos antes de renderizar, un autor de contenido puede controlar además la renderización a través del uso de una política de renderización y/o propiedades de tiempo límite de renderización como se describe a continuación.

40 Política de renderización: progresivas | presentación completa | carga completa  
 Se aplica a: Cuerpo  
 Inicial: progresivo  
 Heredado: no  
 Porcentaje: N/A

[0097] La política de renderización progresiva indica que la visualización puede empezar tan pronto como los recursos esenciales (aquellos marcados como requisitos previos en cabeceras de meta-datos) se han adquiridos. Con esta política, con la cual se adquieren los recursos, se incorporan en los gráficos visualizados y mostrados.

50 [0098] La política de renderización layoutComplete (presentación completa) indica que la imagen renderizada puede no ser visualizada hasta que el software haya adquirido información suficiente para determinar el diseño en pantalla completa y haya adquirido aquellos recursos marcados como requisitos previos. Esta política evita que se tenga la sensación de que los objetos se mueven mientras los gráficos renderizados aparecen en pantalla progresivamente.

55 [0099] La política de renderización loadComplete (carga completa) indica que los gráficos pueden no visualizarse hasta que todos los recursos que se usen para la renderización de la pantalla hayan sido descargados. La única diferencia entre la política de renderización loadComplete (carga completa) y el etiquetado de todos recursos como requisitos previos, es que en el primer caso el evento OnLoad (en carga) tendrá que haber sido entregado al operador apropiado, si lo hay, antes de la renderización, y por lo tanto puede afectar la vista de renderización.

60 [0100] En ciertas circunstancias, la política de renderización especificada puede no ser posible, es decir, si un recurso de requisito previo ha sido eliminado del carrusel y la adquisición a través de un módem ha sido negada por el espectador. En una forma de realización, si no se ha especificado ningún tiempo límite para esta carga, entonces el tiempo límite puede ser el predefinido para un valor indicado (15s) como se muestra en la propiedad de tiempo límite de renderización más abajo. Si se alcanza un tiempo límite, y se han obtenido al menos todos los recursos de requisito previo, lo cual está disponible para la nueva página, se puede visualizar de forma independiente a la política

especificada de renderización. Si algunos de los recursos de requisito previo no se han obtenido, entonces puede ser preferible, si es posible, para la pantalla mostrar la página precedente, si lo hay. Si esto es no posible, entonces puede aparecer un mensaje de error o el descodificador puede renderizar y mostrar aquellos recursos que ha sido capaz de obtener.

5	Tiempo límite de renderización	en el presente   <tiempo>
	Inicial	15s

10 En cualquier caso, mientras el descodificador adquiere los recursos para la nueva página, puede ser preferible continuar mostrando la página antigua y, si es posible, permitir al espectador interactuar con la página antigua.

Transiciones de escena

15 [0101] En una forma de realización, se puede requerir a todos los agentes de usuario que cumplan los dos requisitos siguientes:

- Si el elemento que contiene el vídeo ni se mueve ni cambia de tamaño durante una transición de una página a otra, no habrá ningún problema técnico en el vídeo; y
- si el tamaño o ubicación de un elemento que contiene vídeo realiza algún cambio durante una transición de una página a otra, los cambios en el vídeo y los gráficos se sincronizan estrechamente el uno con el otro.

**3.4 Posición de vídeo y redimensionamiento**

25 [0102] Además de considerar el vídeo como un plano virtual subyacente, el autor de contenido puede colocar cajas de vídeo dentro de contenido HTML usando "broadcast:" como "src", o como fuente de "datos" de un elemento HTML, para cuya ubicación y/o tamaño son especificados. En particular, la ubicación puede ser especificada mediante el uso de CSS.

30 [0103] Los ejemplos siguientes demuestran cómo se puede utilizar un url "broadcast:" en un elemento IMG u OBJECT para solicitar un tamaño de escala en particular.

```
<IMG height=300 width=400 style="position:absolute; left:200px; top:80px" src="broadcast://current ">
<OBJECT height=300 width=400 data="broadcast://current "></OBJECT>
```

35 Los dos ejemplos anteriores requieren que el canal sintonizado habitualmente (identificado por el url "broadcast:") tenga una escala al tamaño de 300 por 400. El primer ejemplo también demuestra cómo las propiedades CSS pueden utilizarse para situar la caja de vídeo resultante. Aunque el tamaño y la posición real del vídeo pueden ser parcialmente determinados por las capacidades del descodificador y los drivers suministrados para el hardware dado, las aplicaciones deberían intentar situar y escalar el vídeo como especificado por el autor de contenido.

**3.5 Soporte para instantáneas MPEG**

45 [0104] Las aplicaciones de HTML también pueden soportar la visualización de imágenes estáticas, tales como MPEG I-Frames, sea en el plano de vídeo o en la capa de gráficos multi-plano. Como los descodificadores frecuentemente poseen hardware especial para una renderización eficaz de MPEG, las imágenes de MPEG son particularmente apropiadas para el entorno televisivo. Las MPEG I-frames se pueden reconocer por el tipo MIME de imagen/mpeg y tendrán una extensión mpg. El siguiente ejemplo demuestra el uso de un MPEG I-Frame.

```
<html>
<head>
<title>example</title>
</head>
<body style="background-image:url(http://pepsi.com/pepsi-
ad.mpg) ">
</body>
</html>
```

**3.6 Control de Audio**

60 [0105] Esta sección trata de reproducir audio de la memoria y controlar la secuencia de audio después de que haya sido seleccionada. Las propiedades auditivas de CSS se pueden utilizar para controlar la secuencia de audio y la reproducción de audio desde la memoria. Las hojas de estilo auditivas permiten a los programadores de contenido controlar el volumen, permiten la presentación de iconos de audio (pistas), e incluso permiten al programador controlar las propiedades espaciales y mezclas. Estas hojas de estilo pueden soportar además las propiedades de volumen, las propiedades de la pausa, y las propiedades de mezcla. El HTML mismo provee una manera de especificar un elemento de audio que usa la marca <object>. Habitualmente hay unos eventos definidos en este elemento: onlayoutcomplete,

onmouseenter, onmouseleave, onreadystatechange.

[0106] Aunque el CSS provee una manera para soportar el control del volumen, un objeto de JavaScript puede utilizarse para ejecutar el "mute". La razón para este requisito es que el objeto necesita recordar el ajuste de volumen previo, de modo que cuando el sonido se restablezca, se ajustará inmediatamente de nuevo al volumen al que estaba el conjunto antes de la reducción del volumen.

#### 4. Obtención de recursos NO-AV

[0107] Las aplicaciones y datos se pueden obtener a partir de fuentes que incluyen la emisión o el punto a punto (p. ej., sobre un canal de retorno a través de módem). En una forma de realización, las aplicaciones HTML pueden proveer acceso para emitir recursos a través de la emisión: protocolo URL, al igual que aquellos que se transmiten dentro de una emisión http: protocolo (bhttp). El acceso a través de la emisión: el protocolo es como se ha descrito anteriormente. Para el protocolo bhttp, cuyo comportamiento del lado del cliente es como se describe abajo, el lado del cliente trata la secuencia de emisión como una caché.

##### 4.1 Acceso a recursos de emisión

###### 4.1.1 Acceso a través de la emisión: protocolo URL

[0108] El programador de contenido HTML/JS puede acceder a recursos de emisión no-AV mediante el uso de la emisión: protocolo de manera similar a la manera que usan la emisión: protocolo para acceder a recursos AV.

###### Una descripción informal del esquema para recursos no-AV

[0109] Aquí, la descripción difiere de aquella provista en la sección precedente en la que se han añadido path\_segments para permitir la especificación de secuencias particulares de datos.

```
broadcast: {/ <service_address> {;i <component-list>}}{/ <path_segments>}
```

Una *service\_address* se define de la siguiente manera:

```
service_address ::= channel_name current
```

donde:

*channel\_name* especifica un nombre estilo DNS que identifica únicamente el canal, y  
*current* especifica el servicio seleccionado habitualmente.

Como se declara en la sección anterior, el *component\_list* es una lista separada por comas que selecciona componentes específicos en la secuencia. El *component\_list* se define de la siguiente manera:

```
component_list ::= component *(", " component)  

component ::= stream_type "=" ( track_tag | "default" | "current" | "none" )  

stream_type ::= "video" | "audio" | "data" | "subtitle" | "teletext"
```

La presencia de path\_segments en un URL indica que se refiere a un módulo específico en el carrusel de datos asociado al *service\_address*. Por ejemplo, el URL "broadcast://tfl.fr/background.png" se refiere al módulo background.png en el carrusel de datos predeterminado.

#### EJEMPLOS

[0110]

broadcast:/background.png cargan el módulo background.png del carrusel de datos predeterminado en el servicio actual.

broadcast://current;data=htp0/Select/broadcast://current;data=htp0/Select el carrusel de datos con track\_tag "htp0", examina el módulo de directorio y carga el módulo "predeterminado" en ese directorio (p. ej., index.htm).

Algunas aplicaciones pueden requerir la capacidad para cargar un módulo específico dentro de un carrusel de datos. Por ejemplo, la siguiente HTML carga el módulo background.png del carrusel predeterminado y lo usa como una imagen de fondo.

```
<BODY background="broadcast:/background.png">
```

Durante una solicitud de carrusel, los eventos típicos HTML que se pueden enviar incluyen.

**Load** (cargar) si la solicitud ha tenido éxito, se envía un evento de carga después de que el URL se haya

cargado finalmente.

**Error** si la solicitud es denegada o de otra manera invalidada se envía un evento de error.

**Abort** (aborto) si el usuario aborta la solicitud antes de que ésta se complete, se envía un evento de aborto.

5 Las aplicaciones residentes (tal como un control de tarea, o EPG) pueden requerir la capacidad de lanzar automáticamente una aplicación durante la selección de servicio.  
En estos casos un URL de la forma

broadcast://cnn.com; data=http0/

10

informa a un navegador para ejecutar automáticamente el módulo predeterminado en un carrusel de datos específico.  
Nota: el módulo predeterminado puede ser seleccionado comprobando el directorio específico para los siguientes módulos. El primer nombre de módulo que existe es cargado automáticamente.

15

BHTTP  
Index.htm  
Index.htm

20 Un URL más simple de la forma "broadcast:/" informa al navegador para que ejecute automáticamente el módulo predeterminado en el carrusel predeterminado del servicio actual seleccionado.

#### 4.1.2 Acceso a través del Http: esquema URL y el carrusel de emisión

25 [0111] En una forma de realización, las páginas HTML pueden usar los URL "http:" para cargar recursos del carrusel. En particular, el caché HTTP se puede mejorar para guardar automáticamente entidades HTTP del carrusel de datos. Por lo tanto, el operador URL http: será capaz de cargar entidades HTTP directamente del caché de HTTP sin abrir una conexión HTTP en el servidor de origen. Por lo tanto, las páginas de HTML que usan un URL "http:" para referirse a entidades HTTP pueden no notar diferencia alguna entre recursos recuperados de la emisión y aquellas recuperadas usando el protocolo URL cliente/servidor HTTP.

30

[0112] Una forma de realización de dicho modelo se ilustra en la Fig. 4. En el ejemplo de la Fig. 4, la cabecera 402 está actuando como un proxy, es responsable de traer datos del servidor de origen 410 que ha sido solicitado por el administrador del carrusel 420 (a través de tantos saltos como se necesite), y para la colocación de las cabeceras de caché apropiadas según sintaxis y semántica HTTP (basada en la cabecera Expires). El descodificador 404 puede entonces poblar su caché desde el carrusel. La entidad de la cabecera Expires puede dar la fecha/tiempo después del cual la respuesta se considera obsoleta.

35

[0113] En respuesta a la detección de un url http, el lado del cliente puede controlar primero su caché local. Si los datos solicitados no se encuentran en la caché, el cliente puede controlar el carrusel actual si lo hay, posiblemente recuperando datos del carrusel. Alternativamente, puede enviar una solicitud HTTP al URL especificado. Para permitir un comportamiento apropiado de la caché, el carrusel puede proveer fechas de expiración y otra información de control caché en cabeceras HTTP. Por ejemplo, tal información puede incluir:

40

1. Información de cabecera de control de caché (HTTP 1.1) que especifica la máxima cantidad de tiempo en que una página particular se puede considerar como actualizada.

45

2. En una respuesta para la cabecera o el cliente, el servidor de origen puede añadir las siguientes cabeceras para permitir un almacenamiento eficaz y preciso:

50

- expires, indicación de la fecha/tiempo después del cual la página se pueden considerar obsoleta;  
- last-modified, indicación de la última vez que los datos fueron modificados en el servidor de origen; y  
- datos ETag (HTTP 1.1), para uso con peticiones condicionales, que proveen un valor indicativo de la página actual (p. ej., algún número de generación o suma de control).

55

3. Peticiones de obtención condicionales que requieren que el descodificador verifique el último valor modificado o el valor ETag resultará en una solicitud apropiada al servidor de origen, que puede devolver el código de estado no modificado si los datos siguen siendo válidos. No obstante, el descodificador se puede configurar para "crear" el tiempo de expiración provisto en una cabecera. Hay que tener en cuenta que el lado del servidor puede modificar el tiempo de expiración real del valor al que fue ajustado por el servidor de origen.

60

[0114] Cabe señalar que debido a que la congestión de red puede retrasar una respuesta, la revalidación de datos que se vuelve obsoleta durante el tránsito podría suponer un bucle infinito. Consecuentemente, HTTP 1.1 especifica que una respuesta puede no ser revalidada para evitar bucles infinitos. Esta regla se puede seguir si los datos vienen del carrusel o directamente del servidor de origen.

65

### 4.1.3 URL relativos

[0115] El uso de URL relativos, que no especifican ni "http:" ni "broadcast:", pueden trabajar con cualquiera de los protocolos. En una forma de realización, los URL relativos pueden ser traducidos automáticamente a uno con el mismo prefijo que fue usado para obtener la página que contenía la referencia. Por lo tanto, si una página se obtuvo usando el URL "broadcast:", luego todas las referencias relativas dentro de esa página también se pueden obtener usando el URL "broadcast:". Porque es posible que las páginas iniciales de una aplicación se puedan descargar a través de "broadcast:", es posible autorizar aplicaciones que nunca especifican explícitamente "broadcast:" o "http:" pero que aún actuarán correctamente.

### 4.2 Control de módem

[0116] En Europa, y en otros lugares, las comunicaciones locales todavía son costosas y puede ser necesario advertir al usuario y quizás mostrar el precio de la comunicación. Mientras que puede depender del sistema realmente el hecho de abrir y cerrar conexiones, puede ser útil para la aplicación notificar al sistema cuándo ha acabado con un sistema. También, en muchas redes, es común que las diferentes aplicaciones requieran conexiones para diferentes números de teléfono, mas que un único número de teléfono asociado a un proveedor de servicio de Internet particular (ISP). En dichos sistemas es común que los diferentes números estén asociados con un único banco de módem con los diferentes números que se usan para contabilizar y otra información. Por lo tanto, la aplicación HTML/JS necesita notificar al sistema cuándo termina de usar una conexión y necesita ser capaz de solicitar una conexión, proveyendo de los parámetros apropiados. Por lo tanto, varias formas de realización pueden soportar los siguientes métodos en el objeto de navegador.modem.

```

navigador.modem.disconnect()
-- indica al sistema que la aplicación
ha terminado de usar la conexión. Puede no haber
eventos asociados a la finalización.
Si una aplicación invoca el siguiente método:
navigador.modem.connect(string parameter, int ms_timeout),

```

el parámetro del string podría contener, por ejemplo, un número de teléfono al que el sistema se pueda conectar. El parámetro ms\_timeout se puede utilizar para indicar durante cuánto tiempo (p. ej., en milisegundos) puede intentar conectarse el sistema. El objeto de "módem" se puede configurar para proveer el estado de conexión como una propiedad sólo de lectura.

[0117] El sistema puede generar automáticamente eventos de conexión cuando algo ocurre en el módem. Algunos ejemplos de dichos eventos de conexión incluyen: success, failure y disconnect\_occurred.

### 4.3 Indicaciones de almacenamiento – requisito previo, enlace y carga previa

[0118] Hay al menos dos indicios importantes que pueden estar presentes dentro de una aplicación HTML para ayudar a la aplicación de HTML/JS del lado de cliente a determinar que recursos tienen prioridad de almacenamiento más alto. Los dos indicios se representan por el requisito previo de meta-datos en la cabecera y el estilo de enlace que se utiliza para indicar qué páginas, aunque no se necesiten inmediatamente, puedan ser solicitadas pronto por la aplicación.

#### Prerrequisito de meta cabecera

[0119] Como se ha explicado anteriormente e ilustrado más abajo, todos los recursos que se marcan como un requisito previo deben estar generalmente disponibles antes de renderizar la página correspondiente para la presentación.

```
<META name="prerequisite" content="http://www.cnn.com/background.mpg">
```

Consecuentemente, los recursos de prerrequisito se pueden identificar y se les ha dado una prioridad más alta para almacenar.

#### Uso de los datos de enlace para precarga

[0120] Además de lo dicho anteriormente, un elemento de enlace, que puede aparecer en la parte <head> de una página, indica los recursos que pueden ser deseados por el espectador de la página actual. Por lo tanto, los recursos catalogados en este elemento son también buenos candidatos para la carga previa en la caché.

[0121] No obstante, se deben observar determinadas advertencias. Por ejemplo, si un documento CSS se cataloga en el elemento de enlace, es posible que se pueda aplicar al documento actual más que a un documento que sería almacenado para uso posterior. Para evitar tal posibilidad, un valor nuevo, carga previa, se introduce para el atributo rel. Si se indica un recurso en una declaración de enlace en la cabecera, y se identifica con una relación de carga previa, entonces el decodificador puede determinar que es un buen candidato para el almacenamiento.

<link rel="prefetch">

**4.4 Un evento que indica que un URL fue actualizado**

[0122] Una de las ventajas de la televisión interactiva es que la presentación del espectador se puede actualizar a tiempo real. Por ejemplo, si hay se marca un gol nuevo en un partido de fútbol, el espectador puede querer recibir una actualización aunque esté viendo un película. Dicha actualización puede emitirse cambiando el contenido correspondiente a un URL. Esta sección describe cómo las aplicaciones pueden ser notificadas cuando el contenido correspondiente a un URL cambia, usando un URLEvent. El objetivo de un URLEvent generado por el agente de usuario se determina por el agente de usuario según las siguientes reglas:

1. Si el URL, cuyo estado ha cambiado, se identifica como el valor del atributo del tipo de nodo correspondiente como se cataloga en la tabla siguiente, entonces el URLEvent es entregado por el agente usuario al nodo correspondiente.
2. Si el URL, cuyo estado ha cambiado, es el url para la página misma, entonces el URLEvent será entregado al cuerpo.

Atributo	Nodo correspondiente
fondo	cuerpo
src	imagen
Datos	objeto
href	enlace
src	Input
src	Frame
src	IFrame

**Atributos**

url de tipo DOMString, sólo lectura

Identifica el URL desde donde el evento fue generado.

**Métodos**

initUrLEvent

El método initUrLEvent se utiliza para iniciar el valor de un URLEvent creado a través de la interfaz DocumentEvent.

Parámetros  
typeArg del tipo DOMString

Especifica el tipo de evento.

canBubbleArg de tipo booleano

Especifica si el evento puede o no crear burbujas.

cancelableArg de tipo booleano

Especifica si la acción predeterminada del evento puede evitarse o no.

urlArg de tipo DOMString

Especifica la url del evento.

Los diferentes tipos de UrLEvents que pueden tener lugar son:

URLInserted

El evento URLInserted tiene lugar cuando un URL se añade al carrusel. Burbujas: si, Cancelable: no , Información de contexto: url

URLUpdated

El evento URLUpdated tiene lugar cuando una versión nueva de un URL se crea en el carrusel. Burbujas: si, Cancelable: si, Información de contexto: url

## URLRemoved

5 El evento URLRemoved tiene lugar cuando un URL es eliminado del carrusel de datos. Burbujas: si, Cancelable: no, Información de contexto: url

10 La acción predeterminada en el caso de URLUpdated (que se puede cancelar llamando un preventDefault()) es recargar el contenido del url asociado). No hay acción predeterminada para URLInserted o URLRemoved. También, cabe señalar que se garantiza que los eventos serán entregados en orden descendente; por lo tanto, si el cuerpo cambia, el evento que representa la actualización del url asociado al cuerpo será entregado antes de entregar cualquier evento que implique URL referido al cuerpo. Hay que tener en cuenta que el evento anterior se puede señalar en el carrusel a través de un directorio delta que indica diferencias entre el último directorio y el directorio actual. De esa manera, una implementación no necesita descargar el contenido entero antes de saber si la app va a ser usada o no, sólo necesita averiguar que hay una nueva versión disponible.

15

### El objeto caché

#### Introducción para indicios de caché dinámica

20 [0123] Ya que típicamente la cantidad de información que se puede presentar en una pantalla es sustancialmente inferior a la contenida en una página que se visualiza típicamente en un PC, un autor que crea contenido para televisión muy a menudo extenderá la misma cantidad de información sobre múltiples páginas. Por lo tanto, el espectador típicamente se desplazará entre páginas y su navegación a través de una página puede ser un buen indicador de los recursos que se necesitarán a continuación. Un autor que hace uso de dicha información transmitiendo indicios basados en esta navegación al agente de usuario puede permitir un rendimiento mucho mejor en los clientes no exclusivos.

25

#### Una interfaz de caché huésped

30 [0124] La interfaz caché soporta dos métodos, carga previa y eliminación. El método de carga previa especifica tanto el URL asociado al recurso que debe ser cargado previamente como una prioridad que indica las posibilidades de que el espectador necesite ese recurso.

35 [0125] El valor de prioridad caché es un número entero no negativo. El autor puede usar un valor de prioridad caché de 0 para indicar que el contenido referido es útil, pero que el autor puede estar inseguro de su probabilidad de uso en la comparación con otras unidades que están solicitando ser almacenadas. El autor puede usar un valor de prioridad caché de 1 para indicar la creencia de que almacenar el recurso específico es muy importante. Un valor muy grande para la prioridad indica que un recurso posiblemente no sea usado (por lo tanto informando al agente usuario que éste puede reclamar la memoria actualmente usada para mantener el recurso asociado del URL en los casos dónde ésta sea necesaria).

40

45 [0126] El método de eliminación se puede utilizar para eliminar una copia almacenada del recurso asociado al argumento URL. Puesto que será eliminado de la caché, y no simplemente invalidado, el sistema no empleará recursos revalidando la entrada. Hay que tener en cuenta que solicitar el método de eliminación es diferente de asignar un número entero muy grande como el valor de prioridad caché en que asignando dicho valor grande de número entero sólo hace que el espacio usado para almacenar este recurso esté más disponible para recoger elementos no utilizados y/o para sostener altos recursos de prioridad.

```
50 interface cache {
    void prefetch(in DOMString URL, in short priority);
    void remove (in DOMString URL);
};
```

#### Unión de la interfaz caché a script

55 [0127] El objeto caché, que implementa la interfaz caché de arriba, es accesible como una propiedad del navegador (navegador: :caché).

#### **Caché de objetos**

60 El **objeto de caché** posee los siguientes métodos:

**prefetch**(URLArg, priorityArg), este método no devuelve un valor.

El URLArg es un tipo de **String**.

El priorityArg es un tipo de **número**.

**Remove**(URLArg), este método no devuelve un valor.

El URLArg es un tipo de **String**.

65

#### El método farPrefetch

[0128] A veces el tamaño de los recursos necesarios para una aplicación dada es muy grande, y, en este caso, normalmente es verdad que muchos de los recursos, por ejemplo, fuentes, en realidad se pueden compartir con otras aplicaciones en diferentes servicios. Cuando se da este caso, los recursos compartidos se reúnen normalmente y se transmiten en un único servicio. Por lo tanto, existe la necesidad de que una aplicación pueda ser capaz de obtener recursos de otro servicio, que normalmente requerirán cambiar temporalmente el sintonizador a una frecuencia diferente y/o al menos elegir un servicio diferente que se soporte en esa frecuencia, almacenado los recursos de ese otro servicio, y sintonizando de nuevo al servicio original. Otro ejemplo de caso para este escenario es el caso donde un espectador quiere descargar correo o información de chat o un juego, y a continuación interactuar con los datos descargados mientras ve un vídeo que se emite en un servicio diferente de los datos descargados.

[0129] En una forma de realización, el siguiente método JS está provisto para permitir que una aplicación sintonice un servicio diferente y descargue información de ese servicio, y luego automáticamente vuelva al servicio original:

```
void navigator.cache.farPrefetch(carouselUrl, ArrayOfUrlsToLoad, functionToCallWhenDone
```

Donde se identifica el URL carrusel a través del protocolo tvx:.

[0130] Las siguientes acciones pueden ocurrir asincrónicamente cuando esta función se cancela. Primeramente, se comprueba el permiso de la aplicación para confirmar que se permite cambiar el servicio. Si esta solicitud es permitida, el servicio específico se sintoniza, todas los URL solicitados se almacenan, luego el sintonizador/demuxer reselecciona el servicio previo, y se solicita la funciónToCallWhenDone. Esta cancelación puede garantizar el hecho de no causar la generación de un evento de interrupción para la aplicación que solicitó el farPrefetch

Evento que define el resultado del método farPrefetch

[0131] El siguiente evento puede aparecer en el objeto de caché después de completar el farPrefetch. El valor detail indica si todos recursos solicitados fueron obtenidos o no. Esto es, en una forma de realización, si se obtiene una menor cantidad que todos los recursos solicitados, entonces se puede considerar que el farPrefetch ha fallado. El autor de contenido debería tener en cuenta que son responsables de solicitar todos los recursos requeridos cuando se utiliza un farPrefetch.

propiedad sólo lectura **detail** es un **número**.

La propiedad **detail** tiene el valor: 1 para éxito,

NaN fallo.

El objeto **CacheEvent** tiene el siguiente método:

```
initCacheEvent(typeArg, canBubbleArg, cancelableArg, detailArg)
```

Este método se utiliza para iniciar el valor de un **CacheEvent** creado a través de la interfaz **DocumentEvent**.

Este método sólo puede ser cancelado antes de que el **CacheEvent** se haya enviado.

Los diferentes tipos de eventos caché que se pueden enviar a **navigator.cache** son:

**FarPrefetchStatus**, este evento notifica que una petición de farPrefetch() ha sido completada.

Burbujas: no

Cancelable: no

Información de contexto: **detail**

**El canal de interacción**

[0132] Las aplicaciones HTML/JS pueden usar el módem(s) fijado a y/o presente dentro de un descodificador para comunicarse con el canal de interacción. Se consideran dos tipos de módem, un módem de encendido permanente (p. ej., cable DOCSIS) y módem convencional (por ejemplo, RTB), cualquiera de los dos o ambos pueden ser accesibles desde un descodificador.

[0133] Dos usos diferentes de canal de interacción han demostrado ser útiles en la televisión interactiva. El uso único, que se encuentra también comúnmente en aplicaciones de PC, es el uso del módem para enviar y/o recibir una cantidad sustancial de datos. Ya que se intercambia una cantidad sustancial de datos, la sobrecarga de establecer una conexión tal como la que está asociada a PPP es insignificante. Sin embargo, un uso diferente ha probado ser una fuente mayor de ingresos para operadores de televisión de pago: la capacidad para llamar un número de teléfono especial, intercambiar opcionalmente unos bytes y colgar. La cantidad de tiempo requerida para establecer un enlace PPP en

este segundo tipo de uso es demasiado y, por lo tanto, indeseable.

[0134] Además del conflicto del uso descrito anteriormente, también es importante el grado de control que una aplicación pueda ejercer sobre una conexión de módem. En una forma de realización, si una aplicación no tiene explícitamente abierto un enlace, la aplicación puede abrir automáticamente un enlace (p. ej., usando un string de conexión que depende de la red), o usar un enlace abierto existente, cuando el acceso de contenido correspondiente a un "http:" url es requerido por la aplicación.

[0135] Para permitir a los desarrolladores ejercitar control sobre protocolos de alto nivel, tal como PPP, la estructura de enlaces descrita abajo puede ser proporcionado. Además, para permitir a las aplicaciones el acceso directo a datos brutos donde los protocolos de nivel alto causan demasiada cabecera, y para permitir aquellas aplicaciones llamar a números de teléfono de premio a través de módem de marcado, la estructura de módem descrita abajo puede ser proporcionada.

### 15 La estructura de enlaces

[0136] La estructura de enlaces definida más adelante se puede usar para (1) controlar explícitamente cuando se abren y se cierran las conexiones, y (2) especificar atributos de conexión. También provee métodos que permiten una aplicación para determinar atributos del enlace.

[0137] Una aplicación de usuario se puede configurar para seleccionar siempre el *mejor* enlace (frecuentemente designado por la red) y especificarlo como enlace predeterminado ([window].navigator.tv.links.default abajo). En tal caso, el autor no siempre necesitará buscar para un enlace con atributos particulares. No obstante, si un autor de la aplicación determina que ésta busca un tipo particular de enlace, pueden acceder directamente al conjunto de enlaces ([window].navigator.tv.links[i] abajo).

Referencia de modelo de objeto:

**[window].navigator.tv.links[i]**  
**[window].navigator.tv.links.default**

[0138] El conjunto de enlaces es una colección de objetos de tipo TVLink tal y como se define más adelante. También, el links.default es del tipo TVLink.

[0139] La propiedad del tipo permite al autor del contenido determinar el tipo de enlace. Mientras los primeros tres tipos se nombran según el protocolo estandarizado que soportan, el cuarto tipo se refiere a un producto particular que soporta un protocolo más ligero en los descodificadores de gama baja.

[0140] La propiedad del estado permite una aplicación para determinar el estado actual del enlace y la propiedad `always_on` (siempre encendido) permite a la aplicación determinar si el enlace es persistente. Si el enlace está conectado y no siempre encendido, la aplicación puede determinar la cantidad de tiempo que el enlace ha sido conectado usando la propiedad temporal.

[0141] Es típico en las redes de televisión de pago que las mismas redes requieran los atributos de conexión sean especificados de una manera formateada de la red. Esto es, una red puede requerir que la aplicación especifique el número de teléfono entero, mientras que otra red sólo permitirá que una aplicación especifique un índice en un conjunto de números de teléfono suministrado por la red, y una tercera red puede no permitir en absoluto la especificación del número de teléfono, pero sólo del nombre de usuario y contraseña. Por lo tanto, el formato de la conexión del atributo de hilo asociado con la petición de conexión depende de la red.

El objeto **TVLink** se define de la siguiente manera.

### Propiedades:

[0142]

tipo	"PPP"   "DVB-RC"   "DOCSIS"   "OTV_Gateway" (sólo lectura)
status	"conectado"   "en conexión"   "desconectado"   "desconectado" (sólo lectura)
siempre encendido	"Verdadero"   "Falso"
tiempo	int -- # segundos conectado (0 si no está en estado "conectado")
nombre	String una propiedad única asociada con este enlace.

Métodos:

[0143]

**open(attributes, timeout) --**

Este método devuelve un número: 1 para OK, -1 si el enlace que fue especificado no existe, -2 si el enlace ya está abierto, y -4 si el permiso para abrir este enlace existente es denegado.

Tener en cuenta que aunque esta llamada puede fallar inmediatamente, la conexión real es asíncrona y se notifica al solicitante a través de un evento LinkUp cuando la conexión ha sido exitosa (o mediante un evento LinkDown si fallara la solicitud).

El parámetro de atributos es String que contiene los atributos de conexión como determinados por el autor de contenido en la consulta con la red (debe conocer al menos formato específico de red).

El parámetro timeout es un número que contiene desconexión por tiempo (en segundos).

close()

Este método devuelve un número: 1 para OK, NaN para fallo

Este método también puede ser asíncronico.

**addEventListener(type, listener, useCapture)**

**removeEventListener(type, listener, use Capture)**

**dispatchEvent(evt)**

-- Estos métodos son los métodos básicos de la interfaz DOM nivel 2 **EventTarget**.

### Eventos:

El **LinkUp** y **LinkDown** son eventos de tipo **LinkEvent**. El objeto **LinkEvent** tiene todas las propiedades de la interfaz de **evento** además de la siguiente propiedad:

la propiedad **detail** de sólo lectura es un **número**.

Donde la propiedad **detail** tiene el valor:

- 1 Para desconexión normal,
- 2 Para cuelgue de línea (por otro lado)
- 3 Ha ocurrido una interrupción,

NaN otro fallo

El **LinkEvent** objeto tiene el siguiente método:

**initLinkEvent(typeArg, canBubbleArg, cancelableArg, detailArg)**

Este método se utiliza para iniciar el valor de un LinkEvent creado a través de la interfaz DocumentEvent.

Este método sólo puede ser cancelado antes de que el LinkEvent se haya enviado.

Los diferentes tipos de eventos de enlace que se pueden enviar a **navigator.modem** son:

#### LinkUp

Este evento notifica que se ha establecido una conexión de módem básico.

- Burbujas : no
- Cancelable : no
- Información de contexto : ninguno

#### LinkDown

Este evento notifica que el módem ha sido desconectado.

- Burbujas: no
- Cancelable: no
- Información de contexto: detail

### La estructura del módem

[0144] La estructura del módem definida abajo se usa para acceder a los datos sin procesar. Por ejemplo, esta estructura es útil cuando una aplicación quiere simplemente marcar un número de teléfono especial, hacer una conexión, y colgar. Esto también puede usarse cuando se necesita intercambiar sólo unos bytes de información, y, en tal situación, los protocolos de nivel más alto requeridos por la estructura de los enlaces de arriba suponen un coste muy elevado para este tipo de uso.

Referencia de modelo de objeto:

**[window].navigator.modem**

5 El objeto de **módem** tiene los siguientes métodos:

**connect(tel, timeout)**

10 Este método devuelve un **número**: 1 para ok, -1 para error de parámetro, -7 módem está en uso, NaN fallo otro.

El parámetro **tel** es de tipo **String** que contiene el número telefónico.

El parámetro **timeout** es un **número** que contiene una desconexión por interrupción (en segundos).

**disconnect()**

15 Este método devuelve un **número**: 1 para OK, NaN para fallo

**sendData(data, timeout)**

20 Este método devuelve un número: 1 para OK, -1 para error de parámetro, -2 no conectado, NaN otro fallo.

El parámetro **data** es un tipo de **String** que contiene una secuencia de valores de byte 0-255.

El parámetro **timeout** es el **número** que contiene una desconexión por interrupción (en segundos).

**receiveData()**

25 Este método devuelve un **String** que contiene los datos disponibles (string vacío si no hay datos disponibles).

**addEventListener(type, listener, useCapture)**

**removeEventListener(type, listener, useCapture)**

30 **dispatchEvent(evt)**

Estos métodos son los métodos básicos de la interfaz DOM de nivel 2 **EventTarget**.

35 El objeto **ModemEvent** tiene todas las propiedades de la interfaz de **evento** además de las siguientes propiedades:

La propiedad **detail** de sólo lectura es un **número**.

La propiedad **detail** tiene el valor: >0 para el número de bytes enviado,

40 - 2 para cuelgue de línea (por otro lado),

- 3 ha ocurrido una interrupción,

NaN otro fallo.

45 El objeto **ModemEvent** tiene el siguiente método:

**initModemEvent(typeArg, canBubbleArg, cancelableArg, detailArg)**

Este método se utiliza para iniciar el valor de un ModemEvent creado a través de la interfaz DocumentEvent. Este método sólo puede ser cancelado antes de que el ModemEvent se haya enviado.

50 Los diferentes tipos de eventos de módem que se pueden despachar a **navigator.modem** son:

**ModemConnect**

55 Este evento notifica que se ha establecido una conexión de módem básico.

- Burbujas: no
- Cancelable: no
- Información de contexto: ninguno

60 **ModemDisconnect**

Este evento notifica que el módem ha sido desconectado.

- Burbujas: no
- Cancelable: no

65

- Información de contexto: detail

La propiedad **detail** tiene el valor:

- 5
- 1 para desconexión normal,
  - 2 para cuelgue de línea (por otro lado),
  - 3 ha ocurrido una interrupción,

NaN otro fallo (p. ej., error de autenticación).

10

### ModemReceiveData

- Información de contexto: detail

15 La propiedad **detail** contiene el número de bytes de datos disponibles para recibir.

### ModemSentData

Este evento notifica que una conexión de módem básico envió algunos datos.

20

- Burbujas: no
- Cancelable: no
- Información de contexto: detail

## 25 5. INTERACCIÓN DE USUARIO

### 5.1 Navegación

#### Foco y resaltado de foco

30

[0145] El CSS2 provee diversas maneras de controlar cómo resaltar elementos resaltados. Por ejemplo, el CSS2 provee tres pseudo-clases relacionadas con la navegación de foco: ":hover", ":active" y ":focus". Además de estas pseudo-clases, el atributo HTML "tabindex" para elementos de entrada y de anclaje también se pueden utilizar para respaldar la navegación. El propósito de este atributo es el de permitir al espectador "tabular" alrededor de la página renderizada antes de seleccionar un elemento. El valor asignado al atributo tabindex determina el orden en el que los elementos serán visitados tras el tabulado.

35

[0146] Determinados estándares de televisión interactiva proveen propiedades "nav-x" para soportar la navegación usando las teclas de dirección (DOM\_VK\_UP, DOM\_VK\_DOWN, DOM\_VK\_LEFT y DOM\_VK\_RIGHT). En particular, ambos DVB MHP y la Asociación de Industrias y Negocios de Radio (ARIB) definen de forma similar, aunque no idéntica, las propiedades "nav-down" "nav-index" "nav-right" "nav-left" "nav-up". En ambas de aquellas especificaciones, la propiedad "nav-index" se utiliza para asociar valores únicos de número entero con elementos particulares de la siguiente manera.

40

```
45 'nav-index'
    valor: <integer none
    inicial: none
    se aplica a: todos los elementos que pueden ser foco
    heredado: no
50 valor de porcentaje: N/A
    tipo medios: tv
```

Porque los elementos con propiedades "nav-index" asociadas tienen asociados valores únicos enteros, el autor de contenido puede usar luego el conjunto de propiedades para controlar la navegación entre elementos.

55

- nav-up
- nav-down
- nav-left, y
- nav-right

60

[0147] Hay varias diferencias entre la definición del DVB-MHP de estas propiedades y la definición provista por la ARIB. El DVB-MHP permite el uso de esta propiedad de controlar la navegación entre marcos permitiendo que el autor de contenido especifique un marco junto a un índice de elementos al que pasar cuando el espectador presiona la tecla de dirección correspondiente. Parece apropiado en receptores de alto nivel el hecho de permitir la navegación entre marcos usando esta propiedad, aunque no está previsto que sean un problema en receptores de tamaño bajo a mediano.

65

[0148] Otra diferencia entre la definición del DVB-MHP de estas propiedades y de la definición asignada por la ARIB es el comportamiento especificado para que ocurra cuando el autor de contenido no provea una o más de estas propiedades para varios elementos. ARIB indica que si una propiedad particular no está especificado para un elemento, el hecho de presionar una tecla de dirección cuando ese elemento está enfocado no ofrece como resultado movimientos del foco. El resultado de aplicar esta regla para elementos para los que ninguna de estas propiedades, excepto el nav-index, hayan sido especificadas es que uno nunca puede navegar fuera de aquellos elementos, si de hecho uno puede navegar hacia los elementos. Adicionalmente, si ninguna propiedad nav-index ha sido especificada para un elemento, entonces no es posible navegar hacia ese elemento. El DVB-MHP especifica un comportamiento predeterminado diferente donde si una de las propiedades no se especifica, entonces la navegación a través de los valores predeterminados de teclas de dirección se predefine al comportamiento del agente usuario.

[0149] En una forma de realización, si la dirección de navegación no se controla explícitamente, el middleware (similar al agente usuario) usa su comportamiento predeterminado para la navegación. Cuando el comportamiento predeterminado no es el comportamiento deseado por el autor de contenido para un movimiento particular, pueden añadir instrucciones para controlar explícitamente la anulación del comportamiento indeseable. De esta manera, no se requiere que los autores de contenido redefinan explícitamente todo el comportamiento que ya encuentran aceptable/deseado. Por lo tanto el comportamiento predeterminado está más estrechamente alineado con el comportamiento de DVB-MHP. La diferencia es permitir la especificación explícita de ambos comportamientos de agente de usuario "default" y "none".

"nav-up"

Valor:	<entero>   ninguno   predeterminado
Inicial:	predeterminado
Se aplica a:	todos elementos que pueden obtener atención
Heredado:	no
Porcentaje:	N/A
Tipo de medio:	tv

"nav-left"

Valor:	<entero   ninguno   predeterminado
Inicial:	predeterminado
Se aplica a:	todos elementos que pueden obtener atención
Heredado:	no
Valores de porcentaje:	N/A
Tipo de medio:	tv

"nav-down"

Valor:	<integer>   none   default
Inicial:	predeterminado
Se aplica a:	todos elementos que pueden obtener atención
Heredado:	no
Valores de porcentaje:	N/A
Tipo de medio:	tv

"nav-right"

Valor:	<integer>   none   default
Inicial:	predeterminado
Se aplica para:	todos elementos que pueden obtener atención
Heredado:	no
Valores de porcentaje:	N/A
Tipo de medio:	tv

Ejemplo de uso:

[0150]

```
<FORM action="http://somesite.com/progladduser" method="post">
<P>
```

```
First name: <INPUT style="nav-index:100; nav-up:105; nav-down:101" type="text" name="firstname"><BR>
Last name: <INPUT style="nav-index:101; nav-up:100; nav-down:102" type="text" name="lastname"><BR>
```

```
email: <INPUT style="nav-index:102; nav-down:103; nav-up:101" type="text" name="email"><BR>
<INPUT style="nav-index: 103; nav-down:104; nav-up: 102" type="radio" name="gender" value="Male">
Male<BR>
<INPUT style="nav-index:104; nav-down: 105; nav-up: 103" type="radio" name="gender" value="Female">
Female<BR>
<INPUT style="nav-index:105; nav-up:104; nav-down:100" type="submit" value="Send"> <INPUT type="reset">
</P>
</FORM>
```

10 Un programador de contenido que requiera control adicional sobre la navegación puede especificar operadores clave usando JavaScript.

### 5.2 Control de teclado virtual

15 [0151] La siguiente propiedad CSS se puede utilizar para el control de la apariencia automatizada de un teclado. Esta propiedad puede ser especificada en una base por elemento para el texto, contraseñas, y elementos de área de texto. Por lo tanto, si una aplicación es consciente de que un elemento de forma particular es un código zip por ejemplo, y por lo tanto la entrada de números a través del control remoto es más fácil, éste puede ser especificado.

	"Teclado virtual"
Valor:	activar   desactivar   auto
Inicial:	activar
aplicable a:	todos los elementos de entrada
Heredado:	no
Valores de porcentaje:	N/A
Tipo de medios:	tv

20 [0152] El valor "disable" significa que el teclado virtual no está disponible cuando el espectador quiere introducir datos en el área, es decir, en cambio puede querer introducir números a través del control remoto. El valor "auto" significa que cuando el elemento al que la propiedad se aplica es el punto de atención, el teclado virtual será presentado automáticamente al espectador. El valor "enable" significa que el teclado virtual será presentado automáticamente al espectador cuando el espectador selecciona el elemento al que se aplica la propiedad. Si las preferencias de usuario del espectador han indicado que hay un teclado alternativo no virtual preferido disponible, entonces el teclado virtual puede no ser visualizado aunque el valor se haya establecido en enable o auto.

30 [0153] Un ejemplo que demuestra como escritores de aplicación podrían prevenir la aparición del teclado virtual para un elemento tipo contraseña es:

```
Input[typ=password]{virtual-keyboard:disable}
```

35 [0154] De forma similar, si la preferencia de usuario indica que el control remoto se puede utilizar como un deletreador numérico, como con un teléfono móvil, entonces ningún teclado virtual aparecerá automáticamente. Alternativamente, el operador de red puede especificar una preferencia de sistema si éste sabe que todos los espectadores tendrán acceso a un teclado físico o a un teléfono móvil.

### 5.3 Entrada por Tecla

40 [0155] Las aplicaciones pueden especificar conjuntos de teclas para los que solicitan notificación por el sistema cuando éstas están en un estado maximizado. Generalmente, aunque no necesariamente, pueden no recibir notificación cuando están en un estado minimizado. La notificación de algunos de los conjuntos de teclas será proporcionada a las aplicaciones solamente en la base en que las hayan solicitado.

45 [0156] No obstante, para otras teclas, la tarea suministrada por la red puede ser "consultada" en cuanto a si la aplicación se puede presentar con las teclas que tiene solicitadas o no. Por lo tanto, es posible que las aplicaciones puedan no ser notificadas de todas las pulsaciones de tecla a las que se hayan suscrito. Las aplicaciones de HTML pueden especificar de que teclas ellos desean recibir notificación estipulando conjuntos de grupos de teclas mostrados en la propiedad de la lista de teclas de abajo. Si el sistema concede la solicitud del grupo de teclas, entonces la notificación del pulsado de tecla se da sólo a la aplicación solicitante y no será enviada a otras aplicaciones (nativas) en el sistema.

50 [0157] Por ejemplo, una aplicación puede conocer que un espectador solo introducirá dígitos entre 1 y 8, aún así quiere ser suficientemente flexible de modo que si el espectador introdujera un 0 o un 9, el canal no cambiará. En este caso, la aplicación puede solicitar la notificación de todas las teclas numéricas, ignorando cualquier cosa excepto los dígitos entre 1 y 8. Es posible que en algunas redes habrá un conjunto predefinido de teclas de que todas las páginas que no lo especifiquen de otra manera, recibirán.

5.4 Lista de propiedades de tecla

[0158] Las aplicaciones tipo HTML pueden añadir una propiedad CSS llamada lista de teclas que indica para que pulsaciones de tecla se puede notificar a una aplicación. Esta propiedad puede ser aplicable al elemento del cuerpo. Un proveedor de contenido que desea más control puede usar el JavaScript apropiado para implementar un control más exhaustivo, haciendo uso del evento on-focus. Todas las páginas usando la misma hoja de estilo compartirán la misma definición de teclas, en que la aplicación está interesada. Esta es una lista separada por comas de grupos de teclas (tal como navegación, selección, información, numérico, color, alfa, etc). Nótese que incluido abajo está el reserved\_set en el valor inicial para lista de teclas aunque estas teclas típicamente están explícitamente marcadas en un típico control remoto. Por lo tanto, aunque estén en el conjunto inicial, puede no haber manera de que un espectador pueda usar estas teclas. Por lo tanto se aconseja a un escritor de aplicación tener cuidado cuando solicita que el espectador pulse estas teclas (p. ej., tener un fallback disponible en el caso de que estas teclas no estén disponibles para un espectador particular.)

"lista de teclas"

Valor <grupo de teclas> + | ninguno

Inicial: Conjunto de desplazamiento, conjunto de navegación, conjunto de selección, conjunto numérico, conjunto de puntuación, conjunto alfa superior, conjunto alfa inferior, conjunto reservado

Aplicable a: body element

Heredado: no

Porcentaje N/A

Valores:

Media type: tv

Donde grupos de teclas pueden ser:

GRUPO DE TECLAS	TECLAS
user_information_set (conjunto de información del usuario)	AYUDA, INFO
scroll_set (conjunto de desplazamiento)	HOME, PAGINA EN ALTO, PÁGINA ABAJO, FINAL
navigation_set (conjunto de navegación)	FLECHA IZQ., FLECHA DERECHA, FLECHA ABAJO, FLECHA ARRIBA
selection_set (conjunto de selección)	CANCEL, ENTER, UNDO
vcr_control_set (conjunto de control vcr)	STOP, PLAY, PAUSE, RECORD and SINGLE_STEP_FORWARD, SINGLE_STEP_REVERSE, FAST_FORWARD, FAST_REVERSE
edition_set (conjunto de edición)	CORTAR, COPIAR, INTRODUCIR
teletext_set (conjunto de teletexto)	MEZCLAR, AGRANDAR, CONTENIDO, MOSTRAR
color_set (conjunto de color)	ROJO, VERDE, AZUL, AMARILLO
numeric_set (conjunto numérico)	0 to 9
Punctuation_set (conjunto de puntuación)	Todos los códigos alfanuméricos en (0x20 to 0x7f)
alpha_upper_set (conjunto alfa superior)	Todos los códigos alfabéticos en (0x41 to 0x5a)
alpha_lower_set (conjunto alfa inferior)	Todos los códigos alfabéticos en (0x61 to 0x7a)
network_set (conjunto de red)	Todos los códigos en (0x0080 to 0x8f)
manufacturer_set (conjunto del fabricante)	Todos los códigos en (0x0090 to 0x97)
Extended_set (conjunto extendido)	Todos los códigos isolations en (0x00a0 to 0xff)
sound_set (conjunto de sonidos)	VOLUMEN BAJO, VOLUMEN ALTO, SIN AUDIO
station_set (conjunto de estación)	CANAL ALTO, CANAL BAJO, CANAL BAJO, CONM RADIO, CONM TV
reserved_set (conjunto reservado)	ETIQUETA, RETROCESO, RETORNO

Ejemplo de uso:

[0159]

<BODY style="key-list: selection\_set, navigation\_set">

5.5 códigos de teclas

[0160] Generalmente hay dos grupos principales de eventos de tecla

[0161] El primero contiene el evento textEvent. El evento textEvent indica que la información de texto ha sido introducido, bien en forma de caracteres imprimibles o en información de texto no imprimible tal como teclas de modificación. Estos eventos textEvent son a veces, pero no necesariamente, acompañados por los eventos de unos segundos grupos mayores de eventos de teclas - keydown y keyup.

## TextEvent

[0162] Este evento indica que ha sido introducida la información del texto. La información de texto introducida puede originarse desde una variedad de fuentes. Podría, por ejemplo, ser un carácter resultante de la pulsación de una tecla. Podría también ser un string resultante de un método de entrada.

[0163] Los eventos keydown y keyup comprenden el segundo grupo de eventos de tecla. Estos eventos se disparan para indicar el movimiento físico de las teclas en el dispositivo de generación de carácter. Dependiendo del sistema de entrada usado, los eventos textEvent pueden o puede que no sean generados por cada par de eventos keydown y keyup.

## keydown

[0164] El evento keydown ocurre cuando se pulsa una tecla.

## keyup

[0165] El evento keyup ocurre cuando se suelta una tecla. Todos estos eventos pueden compartir los siguientes atributos:

TextEvent, keydown, keyup:

burujas: sí

cancelable: sí

información de contexto: 0

context outputString: salida generala por el evento de tecla o nulo.

context keyVal: Carácter Unicode generado por el evento de tecla, ó 0.

context virtkeyVal:

el código virtual generado por el evento de tecla si el evento de tecla no tiene un valor de Unicode, o DOM\_VK UNDEFINED. Aquí está la lista de códigos de teclas virtuales:

```

const unsigned long DOM_VK_UNDEFINED = 0x0;
const unsigned long DOM_VK_RIGHT_ALT = 0x01;
const unsigned long DOM_VK_LEFT_ALT = 0x02;
const unsigned long DOM_VK_LEFT_CONTROL = 0x03;
const unsigned long DOM_VK_RIGHT_CONTROL = 0x04;
const unsigned long DOM_VK_LEFT_SHIFT = 0x05;
const unsigned long DOM_VK_RIGHT_SHIFT = 0x06;
const unsigned long DOM_VK_LEFT_META = 0x07;
const unsigned long DOM_VK_RIGHT_META = 0x08;
const unsigned long DOM_VK_CAPS_LOCK = 0x09;
const unsigned long DOM_VK_DELETE = 0x0A;
const unsigned long DOM_VK_END = 0x0B;
const unsigned long DOM_VK_ENTER = 0x0C;
const unsigned long DOM_VK_ESCAPE = 0x0D;
const unsigned long DOM_VK_HOME = 0x0E;
const unsigned long DOM_VK_INSERT = 0x0F;
const unsigned long DOM_VK_NUM_LOCK = 0x10;
const unsigned long DOM_VK_PAUSE = 0x11;
const unsigned long DOM_VK_PRINTSCREEN = 0x12;
const unsigned long DOM_VK_SCROLL_LOCK = 0x13;
const unsigned long DOM_VK_LEFT = 0x14;
const unsigned long DOM_VK_RIGHT = 0x15;
const unsigned long DOM_VK_UP = 0x16;
const unsigned long DOM_VK_DOWN = 0x17;
const unsigned long DOM_VK_PAGE_DOWN = 0x18;
const unsigned long DOM_VK_PAGE_UP = 0x19;
const unsigned long DOM_VK_F1 = 0x1A;
const unsigned long DOM_VK_F2 = 0x1B;
const unsigned long DOM_VK_F3 = 0x1C;
const unsigned long DOM_VK_F4 = 0x1D;
const unsigned long DOM_VK_F5 = 0x1E;
const unsigned long DOM_VK_F6 = 0x1F;
const unsigned long DOM_VK_F7 = 0x20;
const unsigned long DOM_VK_F8 = 0x21;

```

```

const unsigned long DOM_VK_F9 = 0x22;
const unsigned long DOM_VK_F10 = 0x23;
const unsigned long DOM_VK_F11 = 0x24;
const unsigned long DOM_VK_F12 = 0x25;
5 const unsigned long DOM_VK_F13 = 0x26;
const unsigned long DOM_VK_F14 = 0x27;
const unsigned long DOM_VK_F15 = 0x28;
const unsigned long DOM_VK_F16 = 0x29;
const unsigned long DOM_VK_F17 = 0x2A;
10 const unsigned long DOM_VK_F18 = 0x2B;
const unsigned long DOM_VK_F19 = 0x2C;
const unsigned long DOM_VK_F20 = 0x2D;
const unsigned long DOM_VK_F21 = 0x2E;
const unsigned long DOM_VK_F22 = 0x2F;
15 const unsigned long DOM_VK_F23 = 0x30;
const unsigned long DOM_VK_F24 = 0x31;
const unsigned long DOM_VK_RC_POWER = 0x32;
const unsigned long DOM_VK_RC_TV = 0x33;
const unsigned long DOM_VK_RC_SET_UP = 0x34;
20 const unsigned long DOM_VK_RC_INFO = 0x35;
const unsigned long DOM_VK_RC_RADIO = 0x36;
const unsigned long DOM_VK_RC_NAV = 0x37;
const unsigned long DOM_VK_RC_PIP = 0x38;
const unsigned long DOM_VK_RC_MENU = 0x39;
25 const unsigned long DOM_VK_RC_TEXT = 0x3A;
const unsigned long DOM_VK_RC_HELP = 0x3B;
const unsigned long DOM_VK_RC_SELECT = 0x3C;
const unsigned long DOM_VK_RC_EXIT = 0x3D;
const unsigned long DOM_VK_RC_GUIDE = 0x3E;
30 const unsigned long DOM_VK_RC_RED = 0x3F;
const unsigned long DOM_VK_RC_GREEN = 0x40;
const unsigned long DOM_VK_RC_YELLOW = 0x41;
const unsigned long DOM_VK_RC_BLUE = 0x42;

35 const unsigned long DOM_VK_RC_CHANNEL_UP = 0x43;
const unsigned long DOM_VK_RC_CHANNEL_DOWN = 0x44;
const unsigned long DOM_VK_RC_VOLUME_UP = 0x45;
const unsigned long DOM_VK_RC_VOLUME_DOWN = 0x46;
const unsigned long DOM_VK_RC_MUTE = 0x47;
40 const unsigned long DOM_VK_RC_INFO = 0x48;
const unsigned long DOM_VK_RC_CANCEL = 0x49;
const unsigned long DOM_VK_RC_UNDO = 0x4A;
const unsigned long DOM_VK_RC_STOP = 0x4B;
const unsigned long DOM_VK_RC_PAUSE = 0x4C;
45 const unsigned long DOM_VK_RC_RESUME = 0x4D;
const unsigned long DOM_VK_RC_SINGLE_STEP_FORWARD = 0x4E;
const unsigned long DOM_VK_RC_SINGLE_STEP_REVERSE = 0x4F;
const unsigned long DOM_VK_RC_FAST_FORWARD = 0x50;
const unsigned long DOM_VK_RC_FAST_REVERSE = 0x51;
50 const unsigned long DOM_VK_RC_CUT = 0x52;
const unsigned long DOM_VK_RC_COPY = 0x53;
const unsigned long DOM_VK_RC_PASTE = 0x54;
const unsigned long DOM_VK_RC_MIXING = 0x55;
const unsigned long DOM_VK_RC_MAGNIFY = 0x56;
55 const unsigned long DOM_VK_RC_CONTENT = 0x57;
const unsigned long DOM_VK_RC_REVEAL = 0x58;
const unsigned long DOM_VK_RC_VCR = 0x59;
const unsigned long DOM_VK_RC_SATELLITE_DEL = 0x5A;
const unsigned long DOM_VK_RC_CABLE_DEL = 0x5B;
60 const unsigned long DOM_VK_RC_TERR_DEL = 0x5C;
const unsigned long DOM_VK_RC_DISPLAY_CLOCK = 0x5D;
const unsigned long DOM_VK_RC_SET_CLOCK = 0x5E;
const unsigned long DOM_VK_RC_COLOR_UP = 0x5F;
const unsigned long DOM_VK_RC_COLOR_DOWN = 0x60;
65 const unsigned long DOM_VK_RC_BRIGHT_UP = 0x61;
const unsigned long DOM_VK_RC_BRIGHT_DOWN = 0x62;

```

const unsigned long DOM\_VK\_RC\_CONTRAST\_UP = 0x63;  
 const unsigned long DOM\_VK\_RC\_CONTRAST\_DOWN = 0x64;  
 const unsigned long DOM\_VK\_RC\_PREVIOUS\_CHANNEL = 0x65;  
 const unsigned long DOM\_VK\_RC\_PREFERENCES = 0x66;  
 5 const unsigned long DOM\_VK\_RC\_PARENTAL\_CONTROL = 0x67;  
 const unsigned long DOM\_VK\_RC\_BOX\_OFFICE = 0x68;  
 const unsigned long DOM\_VK\_RC\_PURCHASE = 0x69;  
 const unsigned long DOM\_VK\_RC\_PPV\_SERVICES = 0x6A;  
 const unsigned long DOM\_VK\_RC\_GO\_ONLINE = 0x6B;  
 10 const unsigned long DOM\_VK\_RC\_EXIT\_APP = 0x6C;  
 const unsigned long DOM\_VK\_RC\_SHOW\_INTERACTIVE = 0x6D;  
 const unsigned long DOM\_VK\_RC\_RECORD = 0x6E;

15 contexto inputGenerated:

falso si el evento de tecla no genera ninguna salida visible, tal como el uso de una tecla de función o la combinación de determinadas teclas de modificación usadas en conjunto con otra tecla, verdad si el evento de tecla normalmente causa salida visible. El valor de inputGenerated no garantiza la creación de un carácter, ya que el evento puede ser cancelado.

20 Contexto numPad:

Si fue usado el teclado numérico para generar el evento de tecla, el valor es real, de lo contrario, el valor es falso.

25 Mientras que los códigos anteriores y esta estructura de datos son similares a aquellos definidos en las definiciones de código de tecla DOM-nivel 3, se han añadido códigos para el control remoto. Estos nuevos códigos han sido nombrados DOM\_VK\_RC ... (RC para control remoto). En una forma de realización, las teclas en un teclado que están etiquetadas como si éstas generaran estas teclas. También, se ha declarado arriba DOM\_VK\_HOME en lugar de un RC\_RIGHT, LEFT, RC\_HOME, etc. Otras teclas son posibles y están contempladas.

30 Métodos de evento de tecla

[0166] checkModifier

35 El método checkModifier resulta verdadero o falso, dependiendo de si una única tecla modificadora está asociada a una tecla evento. La lista de teclas abajo representa los parámetros de modificación permitidos para este método.

DOM\_VK\_LEFT\_ALT  
 DOM\_VK\_RIGHT\_ALT  
 DOM\_VK\_LEFT\_CONTROL  
 40 DOM\_VK\_RIGHT\_CONTROL  
 DOM\_VK\_LEFT\_SHIFT  
 DOM\_VK\_RIGHT\_SHIFT  
 DOM\_VK\_META

45 Parametros

tipo de modificador sin firmar con el modificador que el usuario desea solicitar.

Valor de retorno.

Booleano. El estado del modificador representado como un booleano.

50 Sin excepciones.

5.6 Operadores de evento

55 [0167] Además de los oyentes de nivel 2 del modelo de objeto de documento (DOM) , los eventos de tecla se pueden dirigir a operadores de teclas heredadas: onKeyDown, onKeyPress, onKeyUp, plus onFocus, onBlur, onChange y onClick, onSubmit.

6. Seguridad

60 [0168] Dos tipos de seguridad que se pueden requerir en un receptor incluyen:

- (1) protección para recursos de HTML, incluyendo tanto recursos de documento como cookies; y
- (2) acceso protegido a recursos del receptor tales como el sintonizador o módem.

65 Las políticas que rigen la aplicación de los varios mecanismos de seguridad pueden ser establecidas por la red y/o por el fabricante del receptor y los espectadores mismos.

6.1 Protección para recursos de Html

Mismo mecanismo de origen

5 [0169] La misma política de origen se puede definir para restringir una capacidad de recursos para acceder a otros recursos de tal manera que deje al espectador vulnerable. En particular, cuando un recurso intenta acceder una de las propiedades del objeto mostradas en la tabla de abajo, se necesita un control del mismo origen.

10 [0170] En una forma de realización, el primer paso de un control del mismo origen es determinar si el objeto que está siendo referenciado fue creado por el mismo contexto que el script que se ejecuta actualmente. Si es así, el acceso está permitido. De lo contrario, la información adicional se puede examinar para determinar si el url del documento de acceso tiene el mismo origen que el objeto al que se está accediendo. Si el origen es el mismo, entonces el acceso puede estar permitido; de otra manera, el acceso puede ser denegado.

15 [0171] Se puede decir que dos documentos tienen el mismo origen si los siguientes elementos del "protocol://host" (donde el huésped incluye el puerto opcional) son idénticos:

- el protocolo,
- el huésped, y
- 20 • el puerto.

Si cualquiera de estos valores difiere, entonces el acceso puede ser denegado. Se puede presuponer que cualquier dato que se adquiere a través de la emisión: se adquiere en el mismo puerto.

Objeto	Propiedad	Tipo de acceso	Controlado
Ventanas	Todas excepto posición (ver abajo), marcos, padre, y superior	Lectura	Si
	Todas excepto posición (ver abajo)	Escritura	Si
ventana.)posición	Todas	Lectura	Si
	href	Escritura	Si
	protocolo	Escritura	Si
	toString	Método	Si

25 Mecanismo y reglas para cambiar el origen

[0172] Frecuentemente se da el caso de que una sola organización pueda proveer múltiples servidores, pero puede querer permitir ciertos documentos determinados de particulares de estos servidores para acceder a determinados otros documentos proporcionados por servidores distintos de estos. Un mecanismo para permitir tal acción de compartir incluye permitir que un documento cambie su propiedad de dominio (document.) No obstante, tales cambios pueden ser restringidos. Por ejemplo, en una forma de realización esta puede sólo cambiar su dominio a un sufijo apropiado de su dominio actual. Esto es, www.xyz.com se puede cambiar para xiz.com, pero no a abc.com. Adicionalmente, al menos se puede requerir que un periodo permanezca en el nuevo nombre, así, por ejemplo, xiz.com podría no ser abreviado en absoluto. Consecuentemente, si los orígenes de dos recursos diferentes fueron originalmente www.xyz.com e intranet.xiz.com, ambos tendrían que cambiar su dominio para que el acceso estuviera permitido.

40 [0173] Puede haber un problema con el mecanismo para cambiar el origen que se refiere a la internacionalización. El hecho de que de este mecanismo se podría abusar fácilmente en servidores exteriores a los EEUU podría abrir el recurso para todo tipo de ataques a la seguridad. Otro problema potencial es el nivel de detalle de esta regla. Dos recursos del mismo dominio puede que no sean capaces de proporcionarse entre sí acceso sin permitir a otros recursos en ese dominio el mismo acceso. Este problema se puede exacerbar por el mecanismo que permite a los recursos cambiar su dominio.

45 [0174] Una técnica que permitiría un nivel de detalle mayor en compartir usa un mecanismo llamado una credencial. En una forma de realización, una credencial es una declaración firmada por una parte concediendo acceso a uno (o más) de sus recursos a otra parte. La declaración es un trozo de información formateada que identifica al cedente, el cesionario, el recurso a que se concede el acceso, las acciones permitidas en ese recurso (es decir, leer, escribir, u otra propiedad), y opcionalmente una fecha de hasta cuando se concede dicho acceso. La credencial se puede acompañar por una cadena de certificado, el certificado de hoja en la cadena que identifica al cedente y proporciona su clave pública y el certificado de raíz de la cadena que es idéntico a uno de los certificados de raíz en el receptor.

6.2 Acceso en protección a los recursos de receptor.

55 [0175] Las redes frecuentemente prefieren controlar el acceso a ciertos recursos del receptor de hardware y software. Aquellos recursos que puedan ser concedidos para aplicaciones de HTML que se adquieren a través de la emisión se enumeran abajo. El proceso de autorización para conceder estos privilegios para emitir aplicaciones se describe más adelante.

[0176] Mientras que a las aplicaciones que se obtienen directamente de la web se les puede prohibir ejecutar operaciones privilegiadas, una aplicación especial, configurada por o para el operador de red conocido como el UI puede acceder a todas las operaciones privilegiadas de núcleo.

5

[0177] Además de lo dicho anteriormente, a la red se le puede permitir especificar que algunas de las operaciones que siguen abajo puedan admitirse a todas las apps, dondequiera que fueran obtenidas. También, se puede permitir a una red proporcionar parejas de privilegios dominio-nombre.

10 Operaciones privilegiadas de núcleo

[0178] Lo que sigue es una lista de operaciones que pueden estar permitidas sólo cuando el permiso para acceder a ellas está señalado como explicado en la siguiente sección.

15 O descargar módulos desde la emisión

O descargar módulos desde cualquier fuente

O cambiar pistas

O cambiar programas (servicios)

O conectar a un servidor remoto (via telefónica o modem de cable)

20 O hacer cualquier conexión arbitraria

O permitir que algunos módulos no sean firmados (el directorio y los módulos iniciales siempre tienen que estar firmados)

O permitir a la aplicación convertirse en residente en el receptor

O crear o modificar la lista de servicio

25 O usar la lista de servicio

O solicitar que el espectador firme los datos que se suministran para la transmisión

O solicitar que el espectador apruebe el acceso a archivos restringidos y/o números de teléfono

O cambiar algunas preferencias por defecto (exactamente qué preferencias pueden ser modificados depende de los otros privilegios concedidos a la aplicación)

30 O informar al sistema que no necesita (non osd) limpiar la memoria después de la ejecución

O informar al sistema que no necesita limpiar la memoria osd después de la ejecución

O cambiar la ventana de caché EIT

o soltar la caché reservada para EIT

O arbitrar entre solicitudes de eventos de agentes en conflicto

35

Distribución de privilegios de receptor

[0179] En una forma de realización, un módulo de directorio incluye un conjunto de privilegios por-aplicación correspondiente que son solicitados. Este módulo de directorio debe contener una solicitud para este conjunto de privilegios junto con el certificado del productor y debe ser firmado con la clave privada del productor. El certificado del productor se firma usando la clave privada de la red. El certificado del productor declara los privilegios máximos que se pueden conceder a cualquier aplicación bajo ese productor. Por lo tanto, a una aplicación sólo le será concedido un privilegio si está en su conjunto de privilegios por-aplicación y está entre el conjunto de privilegios máximos que se pueden conceder para cualquier aplicación asociada a ese productor. Además de la firma, la seguridad se mejora exigiendo que el directorio firmado contenga un valor hash preciso correspondiente al menos al segmento de código inicial, y opcionalmente a otro código y segmentos de datos usados para la aplicación.

40

45

[0180] Como declarado arriba, todos privilegios de receptor catalogados arriba se pueden conceder al proceso especial conocido como UI. Adicionalmente, los privilegios para aplicaciones recibidas por medio de la emisión se pueden posicionar de la misma manera como se posicionan para aplicaciones de emisión de núcleo. Finalmente, a las aplicaciones recibidas a través del canal de retorno puede que no se les conceda privilegio alguno de receptor. El conjunto de privilegios concedido a una aplicación de emisión o la aplicación UI son conocidas como su conjunto máximo. A menos que la aplicación lo indique de otra manera usando los métodos descritos en la sección siguiente, su conjunto máximo de privilegios es igual al conjunto funcional de privilegios actual. Las aplicaciones pueden establecer su conjunto funcional actual a un subconjunto de su conjunto máximo de privilegios asociados usando los métodos descritos abajo.

50

55

Modo privilegiado mínimo

60

[0181] Usando los métodos descritos en esta sección, una aplicación puede ejecutar en el modo de privilegios mínimos. Este es en realidad un modo mucho más seguro que asegura que antes de usar un privilegio, una aplicación constata específicamente que va a usar ese privilegio. Una ventaja de este modo es que un autor de contenido no puede usar accidentalmente un privilegio que una red concede con demasiada libertad. Usando este modo, por lo tanto, una aplicación no obtiene más privilegios que la red o que permita el receptor (conocido como conjunto máximo), sino más bien manipula cuidadosamente un set de privilegios funcionales que siempre son un subconjunto estricto de ese conjunto máximo.

65

[0182] Para soportar este modo, dos objetos nuevos se requieren en el DOM: (1) el objeto de seguridad y (2) el objeto privilegeManager. Al objeto de seguridad (de clase "Security" ) es accedido a través de la propiedad "security" del objeto global (es decir, el objeto de ventana). El fin del objeto de seguridad habitualmente es contener una propiedad, "privilegeManager", que permite el acceso al objeto privilegeManager (clase "PrivilegeManager" ).

[0183] El objeto privilegeManager tiene cuatro métodos: enablePrivilege, disablePrivilege, revertPrivilege, y removePrivilege.  
Estos métodos permiten un script para manipular privilegios.

enablePrivilege habilita un privilegio para la duración de la función actual.

disablePrivilege deshabilita un privilegio para la duración de la función actual.

revertPrivilege permite a un script revertir un privilegio al estado en el que estaba antes de que se llamara a la función actual

removePrivilege permite a un script eliminar un privilegio de su conjunto máximo. (Este es también eliminado del conjunto de trabajo si está habilitado.)

Cada una de estas funciones resulta en bien un verdadero o falso dependiendo de si la operación fue exitosa. Nota: cuando una función regresa, cualesquiera privilegios habilitados por esa función pueden ser devueltos automáticamente al estado en el que se encontraban en el momento en que se llamó a la función. Cuando script intenta ejecutar una operación privilegiada sin el privilegio necesario habilitado, se ejecutará una excepción TBD apropiada. Si la excepción no se coge, se puede visualizar una caja de diálogo de error antes de abortar el script.

Privilegios adicionales específicos de HTML

[0184] Hay un conjunto de privilegios que son específicos de HTML y principalmente se pueden restringir a un subconjunto de las aplicaciones de emisión HTML. Se puede reservar un conjunto de indicadores para ser usados para operaciones restringidas adicionales.

En una forma de realización, las aplicaciones de HTML pueden usar uno de estos indicadores para indicar si a una aplicación se le concederán todos los privilegios siguientes. (es decir, si el indicador se fija, a la aplicación de emisión HTML le serán concedidas todos los privilegios abajo y si no se fija, a esta aplicación no le serán concedidas todos los privilegios abajo)

O El script puede hacer caso omiso a la misma política de mismo origen, y leer propiedades en otro marco que fue cargado desde un dominio diferente.

O El script puede hacer caso omiso a la política de mismo origen, y cambiar propiedades en otro marco que fue cargada de un dominio diferente.

O El script puede consultar preferencias de usuario desde la aplicación HTML solo un objeto uim.

O El script puede crear, cambiar, y guardar preferencias de usuario desde la aplicación HTML solo objeto uim

O El script puede entregar una forma a mailto: URL

O El script puede manipular cookies cuando y si es añadido un sistema de manejo de cookies más extenso .

O Al script es concedido la unión de los privilegios de extensión de código "runtime" (ejecución) definido tanto en ATSC DASE 1 y DVB HP 1.1.

[0185] Si estos privilegios se conceden a una aplicación de emisión o no puede ser determinado usando el mismo mecanismo descrito en la sección llamada "Posicionamiento de privilegios de receptor". Como mencionado anteriormente, estos privilegios pueden siempre ser concedidos a la aplicación especial UI y/o nunca concedidos a aplicaciones que no son emitidas.

7. Hacia una aproximación declarativa para autoría para SHOWSTOPPERS y prioridades de precarga

[0186] Los primeros lenguajes de programación fueron generalmente muy procedimentales requiriendo un programador para decirle al ordenador cómo llevar a cabo el programa en detalle. Como muestran los ejemplos, la tendencia ha sido hacia lenguajes donde tú especificas lo que hacer, pero no cómo. De estos lenguajes se puede decir que son más declaratorios. En términos generales, un lenguaje declarativo es uno en el que tú especificas qué quieres, y no cómo conseguirlo. Tales lenguajes pueden ser particularmente útiles en el suministro de interfaces de nivel más alto para sistemas complejos subyacentes. Por ejemplo, el HTML puede permitirte especificar qué ha de aparecer en una página, pero no cómo ha de ser ordenado. Otro ejemplo es SQL donde tú especificas qué quieres sacar de una base de datos, pero no dan código para el bucle y las pruebas necesarias para producirlo. Es de notar que la discusión aquí no está estrictamente limitada a lenguajes declarativos per se. Más bien, HTML, JavaScript, CSS, y otros constructos y lenguajes de este tipo y son contemplados. En una forma de realización, son contemplados lenguajes y constructos que son comúnmente usados en la creación y manipulación de contenido de web . En cualquiera de estos casos, las declaraciones u otras instrucciones usadas en la creación y/o manipulación de recursos y contenido en este documento

pueden ser generalmente referidos como "instrucciones".

#### Antecedentes

5 [0187] Esta sección (1) describe el showstopper y requisitos de precarga; (2) identifica como tal información puede ser llevado a cabo tanto en DASE como en DVB-MHP; y (3) propone maneras en que los autores pueden indicar ambos showstopper y recursos de precarga dentro de sus documentos XHTML.

10 [0188] Aunque los detalles de una implementación de transcodificado no están descritos, los expertos en la técnica pueden averiguar que los valores iniciales asignados a showstopper y recursos de precarga pueden ser trasladados automáticamente a instalaciones para transporte existentes de DASE/DVB-MHP.

#### Showstopper y requisitos de precarga

15 [0189] Los creadores de contenido frecuentemente desean usar recursos múltiples en la construcción o presentación de una escena y pueden considerar la adquisición de un subconjunto de estos recursos esenciales antes de la visualización al espectador. Esto es, pueden preferir que la escena antigua debería continuar siendo visualizada hasta al menos los recursos esenciales han sido recibidos y descodificados. Estos recursos esenciales se pueden designar showstoppers porque los creadores no quieren visualizar algo hasta que al menos estos recursos esenciales estén disponibles. Además, si estos recursos nunca se hacen disponibles, los creadores de contenido pueden preferir que o sea visualizado nada. Además, el marcado de estos recursos como esenciales puede permitir que la secuencia de emisión sea empaquetada más fácilmente para mejorar el rendimiento.

20 [0190] En general, se puede mejorar el rendimiento por una precarga inteligente de recursos. En particular, pueden ser posibles destacables mejoras de rendimiento cuando las prioridades en precarga pueden ser modificados dinámicamente dependiendo de la interacción del espectador. Por lo tanto, es deseable permitir a los autores de contenido estipular tanto los recursos esenciales así como la priorización de precarga (dinámicamente modificable).

25 [0191] La Fig. 5 ilustra una forma de realización de un método para precargar recursos de prerrequisito. En el ejemplo mostrado, un proxy centralmente ubicado ejecuta un preprocesado o transcodificación de contenido que es solicitado por un cliente o de otra manera destinado a un cliente. Cuando el proxy recibe el contenido incluyendo instrucciones de presentación (bloque 502), el proxy puede escanear el contenido para instrucciones que indican que cierto contenido es considerado un prerrequisito para la presentación. Si tales instrucciones no son detectadas (bloque de decisión 504), entonces las instrucciones (o señales y/o datos correspondientes a las instrucciones de presentación) se envían al cliente (bloque 16) y la presentación puede iniciarse (bloque 518).

30 [0192] Por otro lado, si tales instrucciones de prerrequisito se detectan por el proxy, el proxy puede inmediatamente transportar una indicación al cliente (bloque 506) de que estos recursos identificados son considerados prerrequisitos. Tras la recepción de esta indicación, el cliente puede entonces determinar si tiene o no actualmente los recursos de prerrequisito identificados (bloque de decisión 508). Si el cliente no tiene estos recursos, el cliente entonces puede emprender cualquier acción necesaria para la precarga de los recursos de prerrequisito (bloque 510). Posteriormente, o al mismo tiempo, el proxy puede transportar el contenido de presentación restante o instrucciones al cliente (bloque 512). Una vez el cliente ha obtenido los recursos de prerrequisito (bloque de decisión 514), está permitida la presentación del contenido correspondiente a los recursos de prerrequisito.

35 [0193] Debe entenderse que son posibles numerosas alternativas. Por ejemplo, en una forma de realización alternativa, no hay un proxy como se describe. Más bien, el cliente se configura para procesar recursos y contenidos directamente. En tal forma de realización, el cliente se puede configurar para primero escanear el contenido recibido para instrucciones de prerrequisito. Alternativamente, las instrucciones de prerrequisito se pueden procesar conforme se reciben. Otras formas de realización son posibles y están contempladas.

#### Soporte dentro de DASE y DVB MHP

40 [0194] Actualmente ni la DASE de la DAE ni DVB-HYML de MHP proporcionan una instalación que permita a los autores de contenido identificar showstoppers o priorizaciones de precarga. No obstante, proporcionan instalaciones que pueden ser usados para transmitir dicha información.

#### Soporte dentro de DASE

45 [0195] Hay soporte explícito para la identificación de la prioridad estática inicial de recursos dentro de una aplicación en la entidad de raíz del DASE DTD. Este soporte está en forma de la definición de un valor de prioridad para un artículo de caché que se une con un atributo de precarga. Quizás en el nivel DASE 2, para mejorar la entidad de raíz DTD de modo que esta incluya soporte para identificación de showstopper; esto es, una forma de realización posible sería añadir un atributo llamado showstopper.

50 [0196] Antes de tal adición, por supuesto, DDE-2 podría recomendar el uso de x-dde2-showstopper como un valor de

atributo no estandarizado. Los elementos que identifican los showstopper y prioridades de precarga inicial podrían ser automáticamente formuladas desde las mejoras de HTML propuestas en la siguiente sección y, por lo tanto, estarían disponibles al receptor tan pronto como la aplicación entre el estado iniciado. No son necesarios para modificar las prioridades de precarga en el elemento de raíz en respuesta a la interacción del usuario, así que esta mejora de menor entidad, con la propuesta del autor abajo, bastaría para soportar completamente los requisitos de precarga y showstopper en el DASE DAE.

Soporte dentro de DVB-MHP

[0197] DVB-MHP proporciona un descriptor opcional, conocido como el descriptor de precarga, dentro del AIT. Como con el atributo de precarga del elemento de raíz DASE, este descriptor podría ser automáticamente generado en las mejoras de HTML propuestas abajo. Los recursos de showstopper podrían ser colocados de una de las distintas maneras; bien añadiendo un nuevo descriptor AIT para recursos de showstopper o, alternativamente, o ajustando la prioridad de los recursos de showstopper al valor más alto posible (100)

Propuesta de autoría

SHOWSTOPPERS

[0198] Los autores de contenido pueden desear que exista una manera de identificar aquellos recursos de manera que si no son obtenidos por un receptor, la visualización debería ser retrasada.

Propuesta mínima

[0199] Se propone que DDE pueda definir un perfil para pares de nombre/valor META específico de DDE. Entre aquellos pares estaría el nombre "prerequisite," con el valor siendo el objetivo URI del recurso esencial. Un ejemplo de este par de nombre/valor sería el primero abajo que indica que "background.mpg" es un recurso esencial que necesita ser adquirido y procesado por el receptor antes de la visualización del contenido inicial de la aplicación.

```
<META name="prerequisite" content="http://www.cnn.com/background.mpg">
```

Priorización de precarga

[0200] Como se ha mencionado antes, los autores de contenido pueden querer proporcionar un indicio en lo que se refiere tanto a parámetros de emisión como a comportamiento de almacenamiento indicando que puede ser deseable inicialmente precargar ciertos recursos, independientemente de si aquellos recursos se consideran como esenciales o recursos de prerequisite tal y como se ha definido anteriormente. No es necesario que la priorización de precarga inicial suministrada por el autor sea idéntica a la priorización que eventualmente es soportada en el fichero de señalización correspondiente (es decir, el elemento de raíz DASE o el descriptor de precarga MHP). No obstante, los desarrolladores de contenido típicamente no son buenos eligiendo entre demasiadas prioridades diferentes. (Prioridades numéricas absolutas, tal como un valor entre 1 y 100 son frecuentemente mejor elegidas por métricas más complicadas que responden al tamaño del recurso, tamaño previsto de cache, índice de transmisión de la secuencia de emisión, etc.)

[0201] Por lo tanto, como propuesto aquí, se puede permitir al autor de contenido identificar si es o no deseable para un receptor precargar un recurso en particular. Por ejemplo, el autor de contenido puede identificar recursos para ser precargados usando el elemento de enlace en el <head> (cabecera) del documento inicial y por definición de un nuevo valor "prefetch" (precarga) para el rel atributo de este elemento. Ya que puede haber diferentes recursos que el autor quisiera recomendar para precargar, puede indicar una prioridad de precarga también. Por ejemplo, puede ordenar estos recursos múltiples de modo que los primeros tengan prioridad más alta que los últimos.

[0202] Como el DOM permite la modificación dinámica de la lista de recursos de enlace a tiempo de ejecución, por ejemplo, basado en la interacción del usuario, los recursos de enlace modificados pueden servir como indicio al receptor en lo que se refiere a prioridades dinámicamente cambiantes. No obstante, también puede ser útil permitir al autor de contenido no sólo controlar prioridades de precarga dinámicamente cambiantes, sino también indicar que el uso de un recurso es inminente de modo que el terminal puede desear "precrear" el recurso (p. ej., asignar recursos tales como memoria, y decodificación) en vez de simplemente precargar el recurso. Para permitir al autor de contenido realizar esto, se puede utilizar un objeto caché que implemente tanto un método prefetch() (precarga) al igual que precreate() (precreación).

8. Identificadores de recursos uniformemente extendidos para emisiones televisivas

[0203] El uso de estándares W3C para el contenido de televisión interactiva de autor que debe ser soportado con señales de televisión digital ha comenzado a aumentar significativamente. RFC 2838 (identificadores de recurso constante para emisiones televisivas) fijó su objetivo en la necesidad de referirse a secuencias de emisión televisiva como un conjunto; esta sección extiende la descripción contenida aquí para incluir la capacidad para dar referencia a subsecuencias particulares y recursos sin vídeo que también se pueden soportar la secuencia de emisión. Además de

ser útil directamente dentro del descodificador del cliente existente o aplicaciones televisivas, el esquema descrito aquí se puede aplicar a esquemas de televisión propuestos específicos de transporte, por ejemplo, dvb, ocap, y arib. El propósito de tal mapeo es la de permitir que un programador de contenido autorice su contenido usando el uri descrito aquí, mientras se permite una transcodificación automática (o manual) a uno o varios de los otros esquemas propuestos.

Identificador de recurso de uniforme de televisión extendida (URI)

[0204] La estructura básica de la televisión extendida uri es:

tvx:<service-address>[<track-list>][<abs-path>]

donde

<service-address> es una descripción del generador de señales, que puede corresponder a identificadores estilo DNS definidos para "tv:" en RFC 2838.

El opcional

<track-list> puede especificar, audio, vídeo, subtítulos, teletexto, o subsecuencias de datos en la secuencia que emana del servicio- adress. El

<abs-path> puede utilizarse para identificar recursos individuales dentro de una subsecuencia, o, puesto que su sintaxis es bastante flexible, puede ser además definido por varios de los transportes-especificos URIs.

Canal actual

[0205] El canal actual puede ser especificado como

tvx://current

[0206] Este URI se refiere a cualquier emisión televisiva a la que está siendo accedido por el objeto de referencia. Esta definición difiere de la definición "tv:", ya que es específica del objeto de referencia. Esta diferencia es necesaria porque los descodificadores que contienen sincronizadores múltiples, decodificadores, etc. están haciéndose más habituales.

[0207] Este emisión "actual" puede contener múltiples audios (p. ej., idiomas diferentes), múltiples vídeos (p. ej., ángulos de cámara diferente), y diferentes tipos de datos. No obstante, este URI se refiere a sólo aquellas subsecuencias que se usan para la destinación asociada al objeto de referencia. Por ejemplo, si hay subtítulos tanto alemanes como ingleses disponibles, pero la pantalla asociada al objeto que se refiere a este URI sólo muestra los subtítulos en alemán (es decir, no muestra los subtítulos en inglés), entonces, los subtítulos en inglés no serían parte de tvx://current.

Sintaxis (BNF) para URIs extendidos de televisión

[0208] Lo siguiente es un ejemplo de una especificación formal para la televisión extendida URIs:

```

tvx_uri = "tvx:" [tvx_hier_part]
tvx_hier_part = tvx_net_path | tvx_abs_path
tvx_net_path = "/" service_addr [comp_list] [tvx_abs_path]
service_addr = broadcast | "current"
comp_list = ";" component *("," component )
component = stream_selector
stream_selector = stream_type "=" stream_id
stream_type = "video" | "audio" | "data" | "subtitle" | "teletext"
stream_id = 1*a1phanum | "default" | "current" | "none"
tvx_abs_path = "/" path_segments
    
```

where:

broadcast puede ser definido como RFC2838)  
 path\_segments pueden ser definidos en RFC 2396  
 alphanum puede ser definido en RFC 2396

Semántica para URIs extendidos de televisión

[0209] Esta sección define el significado de las varias formas de los URIs de televisión extendidos.

Dirección de solo servicio

[0210] La subsecuencia referenciada por una dirección de solo servicio puede consistir en vídeo, audio, teletexto, subtítulos, y secuencias de datos. Las secuencias de datos pueden contener código ejecutable además de datos

usados por el código o datos usados por una aplicación de residente. Además, puede haber más de una secuencia de cada tipo en la subsecuencia referenciada. Por ejemplo, `tvx: //bcd.com` puede contener 2 secuencias de vídeo, 4 secuencias audio, una secuencia de teletexto, una secuencia de subtítulos, y cinco secuencias de datos. Que secuencia son "mostradas" por el objeto que hace referencia a este URI puede depender de muchos factores. Si el espectador tiene seleccionado un ajuste predeterminado que indica una preferencia en lo que se refiere a si son visualizados teletexto y/o subtítulos o no, entonces esa preferencia se puede utilizar para determinar si estas secuencias son visualizadas. Adicionalmente un espectador puede indicar un lenguaje de audio preferido.

[0211] La red de emisión puede usar señalización para indicar el flujo de vídeo predeterminado, y, por ejemplo, en el caso de DVB MHP, puede indicar que aplicaciones particulares deberían ser descargadas y ejecutadas. Si el receptor tiene la capacidad de descodificar al menos una secuencia de vídeo y una secuencia audio al mismo tiempo, entonces en una forma de realización al menos una de estas será descodificada cuando un `tvx uri` de esta forma es especificado. Además, se pueden proporcionar al espectador controles que le permitan "silenciar" el audio o vídeo. Si el espectador no tiene silenciada una secuencia, pero tampoco tiene seleccionado una preferencia, y la red no tiene indicada una preferencia, entonces cualquiera de las secuencias correspondientes se pueden descodificar y visualizar.

[0212] Como declarado arriba, mientras también puede ser usado un URI de la forma `tvx://current` hacer referencia a este URI no cambia generalmente qué flujos son descodificados (y presentados).

Especificando componentes

[0213] El autor de contenido puede hacer referencia a subsecuencias particulares en el flujo usando este URI. Por ejemplo, `tvx: //bcd.com; audio=eng` puede referirse a una subsecuencia de audio en inglés. También, se puede hacer referencia a más de una secuencia usando esta forma de URI. Por ejemplo, `tvx://bcd.com; video=catcher; audio=eng` se puede utilizar para referirse a un vídeo que está grabado detrás de un receptor de béisbol junto con el audio en inglés. Se espera que el autor de contenido pueda tener herramientas apropiadas por las que puede ajustar o bien un "track tag" (p. ej., receptor, eng) para corresponder a una subsecuencia en particular, o para que un conjunto track tags se pueda determinar por un productor estándar o de vídeo, por ejemplo.

[0214] En una forma de realización, hay dos claves especiales que se pueden utilizar como etiquetas de la pista que se definen en este documento: "actual" y "predeterminado." La etiqueta de la pista "actual" indica la subsecuencia que está siendo visualizada actualmente. Por ejemplo, si el espectador está viendo actualmente una película y está escuchando al audio francés, su audio se puede cambiar al inglés sin afectar el vídeo a través del uso del siguiente URI: `tvx://current; video=current; audio=eng` (siempre que la etiqueta de pista "eng" haya sido asociada al audio).

[0215] La clave "por defecto" se puede utilizar para referirse al fichero tal y como se define por el espectador, autor, receptor, autor de contenido o alguna combinación, según una particularización y/o especificación particular. Esto es, en algunas redes verticales, el operador de red puede tener la autoridad para ajustar una preferencia por defecto y en otras redes, puede depender del espectador.

Segmentos de Path

[0216] Los segmentos de path se pueden utilizar para identificar un recurso dentro de un componente particular. Por ejemplo, `tvx: //bcd.com; data=novice/game/chess/move3` puede referirse al recurso `game/chess/move3` que se soporta en la subsecuencia de datos con la etiqueta de pista de principiante.

[0217] Se pueden asignar significados adicionales X a los segmentos del path cuando los URI de televisión específicos de transporte se mapean a este URI. No obstante, hasta que sean definidos así, los segmentos path sólo serán significativos cuando el tipo de componentes sean datos.

[0218] Varias formas de realización pueden incluir además de recepción, envío o almacenamiento de instrucciones y/o datos implementados conforme a la descripción precedente sobre un medio portador. En términos generales, un medio portador puede incluir medios de transmisión o señales usados en sistemas emisión y de otra manera tales como señales eléctricas, electromagnéticas o digitales, transmitidas a través de un medio de comunicación tal como red y/o un enlace inalámbrico. Por ejemplo, un operador de red puede transmitir señales que describen instrucciones de programa a través de un sistema de emisión. Un medio portador también puede incluir medios de almacenamiento o medios de memoria tales como medios ópticos o magnéticos, por ejemplo, disco o CD-ROM, medios no volátiles o volátiles tal como RAM (p. ej. SDRAM, RDRAM, SRAM, etc.), ROM, etc.

[0219] Debe entenderse que las formas de realización anteriores se destinan a ser sólo ilustrativas. Numerosas configuraciones alternativas son posibles y están contempladas.

## REIVINDICACIONES

- 5 1. Método para presentación en un dispositivo de cliente (30, 1002,404), comprendiendo el método:  
 recepción de una o más instrucciones por un dispositivo (21) que comprende un medio de transmisión para transmitir  
 señales al dispositivo de cliente (30,1002,404), donde dichas instrucciones se proveen en una forma de información  
 textual para crear y/o controlar una presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos;  
 10 determinación por dicho dispositivo (21) si una o más instrucciones incluyen una instrucción prerequisite  
 que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerequisite para iniciar la  
 presentación (504);  
 iniciación de dicha presentación mediante una unidad de control (1030), enlazada al dispositivo de cliente, antes de  
 recibir todos los conjuntos de recursos, en respuesta a determinar que una o más instrucciones no incluyen dicha  
 15 instrucción prerequisite (518); y,  
 en respuesta a la determinación de que una o más instrucciones incluyen dicha instrucción prerequisite (514),  
 determinando mediante la unidad de control (1030) si el dispositivo de cliente (30, 1002, 404) tiene dicho subconjunto de  
 recursos y prohibiendo el inicio de dicha presentación hasta que sea adquirido dicho subconjunto de recursos,  
 que comprende además mejorar una entidad de raíz en el DTD según ATSC DASE para añadir un atributo  
 20 showstopper, comprendiendo dicho atributo showstopper una indicación de que la adquisición de recursos particulares  
 del conjunto de recursos son un prerequisite para iniciar dicha presentación.
2. Método según la reivindicación 1, que comprende además el dispositivo de cliente (30, 1002,404) que recibe una  
 indicación que dicho subconjunto de recursos es un prerequisite para la iniciación de la presentación con anticipación a  
 la recepción de los recursos restantes del conjunto de recursos.
- 25 3. Método según la reivindicación 2, donde la recepción de dicha una o más instrucciones es por un servidor (21), y  
 donde el método comprende además el servidor (21) que escanea una o más instrucciones para determinar si está  
 presente una o más instrucciones prerequisite que identifican que el subconjunto de recursos está presente, y si está  
 detectada la instrucción prerequisite (504), enviando el servidor (21) una indicación al dispositivo de cliente (30,  
 30 1002,404) de que dicho subconjunto de recursos son considerados prerequisites (506).
4. Método según la reivindicación 3, donde el servidor (21) transporta dicha indicación al dispositivo de cliente (30,  
 1002,404) antes de transmitir otras instrucciones o señales y/o datos correspondientes a una o más instrucciones al  
 dispositivo de cliente (30, 1002,404).
- 35 5. Método según la reivindicación 1, donde dicha instrucción prerequisite es seleccionada del grupo que consiste en: un  
 idioma de marcado, un idioma de escritura, y una hoja de estilo.
6. Método según la reivindicación 5, donde una o más instrucciones mencionadas se reciben por un servidor proxy (21)  
 40 en un sistema de televisión interactivo (100).
7. Método según la reivindicación 6, donde dicha determinación se realiza por dicho servidor proxy (21), y donde dicho  
 método comprende además dicho servidor proxy (21) que transmite señales que comprenden una indicación que  
 45 identifica dicho subconjunto de recursos a un dispositivo de cliente remoto (506).
8. Método según la reivindicación 7, que comprende además dicho dispositivo de cliente (30, 1002,404) que adquiere  
 dicho subconjunto de recursos en respuesta a la detección de dichas señales (510).
9. Método según la reivindicación 8, donde dicho subconjunto comprende secuencia de audio y/o vídeo, y donde la  
 50 adquisición de la secuencia de audio y/o vídeo comprende la configuración de recursos de hardware dentro de dicho  
 dispositivo de cliente (30, 1002, 404).
10. Método según la reivindicación 8, donde la adquisición del subconjunto de recursos comprende el dispositivo de  
 55 cliente (30, 1002, 404) inicia peticiones para recursos localizados remotamente para ser conducidos a dicho dispositivo  
 de cliente (30, 1002, 404).
11. Método según la reivindicación 1, donde dicha prohibición es además otra respuesta para detectar que un tiempo  
 correspondiente de expiración no ha caducado aún, y donde dicho método comprende además permitir la presentación  
 60 de dicha presentación en respuesta a la detección de que dicho tiempo para la expiración ha caducado.
12. Sistema de televisión interactivo (100) que comprende un dispositivo de cliente (30, 1012, 404) y un servidor proxy  
 remoto (21) configurado para recibir una o más instrucciones para crear y/o controlar en el dispositivo de cliente, una  
 presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos, **caracterizado por el hecho de que** dichas  
 65 instrucciones se proveen en una forma de información textual y  
 dicho servidor proxy remoto (21) es además configurado para:  
 determinar si dichas una o más instrucciones incluyen una instrucción prerequisite que indica que la adquisición de un

subconjunto de dicho conjunto de recursos es un prerrequisito para la iniciación de la presentación (504); transmitir una o varias primeras señales al dispositivo de cliente remoto (30, 1012, 404), en respuesta a la determinación de que una o más instrucciones incluyen dicha instrucción prerrequisito (506); comprendiendo dichas una o varias primeras señales una indicación que identifica dicho subconjunto de recursos; y

5 transmitir una o varias segundas señales al dispositivo de cliente remoto (30, 1012, 404); comprendiendo dichas una o varias segundas señales una o más instrucciones mencionadas;

dicho dispositivo de cliente (30,1012, 404) está configurado para:

10 recibir dichas una o varias primeras señales;  
 recibir dichas una o varias segundas señales; e  
 iniciar dicha presentación antes de la recepción de todos los dichos conjuntos de recursos en el caso de que ningún subconjunto de recursos haya sido identificado,  
 de lo contrario, determinar si el dispositivo de cliente tiene dicho subconjunto de recursos (508) y prohibir la iniciación de  
 15 dicha presentación hasta que dicho subconjunto de recursos sea adquirido,  
 que comprende además una mejora de una entidad de raíz en el DTD según ATSC DASE para añadir un atributo showstopper, comprendiendo dicho atributo showstopper una indicación de que la adquisición de los recursos particulares del conjunto de recursos son un prerrequisito para iniciar dicha presentación.

20 13. Sistema según la reivindicación 12, donde dicha instrucción prerrequisito es seleccionada del grupo que consiste en: un idioma de marcado, un idioma de escritura, y una hoja de estilo.

14. Sistema según la reivindicación 13, donde la adquisición de dicho subconjunto de recursos comprende dicho dispositivo de cliente (30, 1012,404) que configura recursos de hardware dentro de dicho dispositivo de cliente.

25 15. Sistema según la reivindicación 13, donde la adquisición de dicho subconjunto de recursos comprende iniciar peticiones para recursos remotamente localizados para ser conducidos a dicho dispositivo de cliente (30, 1012,404).

30 16. Sistema según la reivindicación 12, donde dicho dispositivo de cliente (30, 1012,404) se configura para prohibir dicha iniciación en respuesta a detectar que un tiempo correspondiente para la expiración no ha expirado aún, y donde dicho dispositivo es posteriormente configurado para permitir la presentación de dicha presentación en respuesta a la detección de que dicho tiempo para la expiración ha expirado.

35 17. Dispositivo de cliente (30, 1012,404) en un sistema de televisión interactivo (100), comprendiendo dicho dispositivo de cliente un receptor (1026) y una unidad de procesamiento (1030) acoplada a dicho receptor para crear y/o controlar una presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos, **caracterizado por el hecho de que:**

dicho receptor (1026) se configura para recibir señales correspondientes a instrucciones proporcionadas en una forma de información textual; y

40 dicha unidad de procesamiento (1030) es configurada para:

determinar si una o más instrucciones incluyen una instrucción prerrequisito que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerrequisito para la iniciación de la presentación (504);  
 45 iniciar dicha presentación antes de recibir todos los dichos conjuntos de recursos, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerrequisito  
 en respuesta a determinar que una o más instrucciones incluyen dicha instrucción prerrequisito (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos y prohibir la iniciación de dicha presentación hasta que se haya adquirido dicho subconjunto de recursos,  
 que comprende además una mejora de una entidad de raíz en el DTD según ATSC DASE para añadir un atributo  
 50 showstopper, comprendiendo dicho atributo showstopper una indicación de la que la adquisición de recursos particulares del conjunto de recursos son un prerrequisito para iniciar dicha presentación.

18. Medio portador que comprende instrucciones de programa ejecutable para:

55 recibir instrucciones que se proporcionan en una forma de información textual para crear y/o controlar una presentación audio, vídeo y/o gráfica que requiere un conjunto de recursos;

determinar si una o más de dichas instrucciones incluyen una instrucción prerrequisito que indica que la adquisición de un subconjunto de dicho conjunto de recursos es un prerrequisito para la presentación (504) por un cliente;

60 iniciar dicha presentación antes de la recepción de todos los conjuntos de recursos mencionados, en respuesta a determinar que una o más instrucciones no incluyen dicha instrucción prerrequisito (518); y,

en respuesta a determinar si una o más instrucciones incluyen dicha instrucción prerrequisito (514), determinar si el dispositivo de cliente tiene dicho subconjunto de recursos y prohibir la iniciación de dicha presentación hasta que dicho subconjunto de recursos sea adquirido,

65 que comprende además una mejora de una entidad de raíz en el DTD según ATSC DASE para añadir un atributo showstopper, comprendiendo dicho atributo showstopper una indicación de que la adquisición de recursos particulares del conjunto de recursos es un prerrequisito para iniciar dicha presentación.

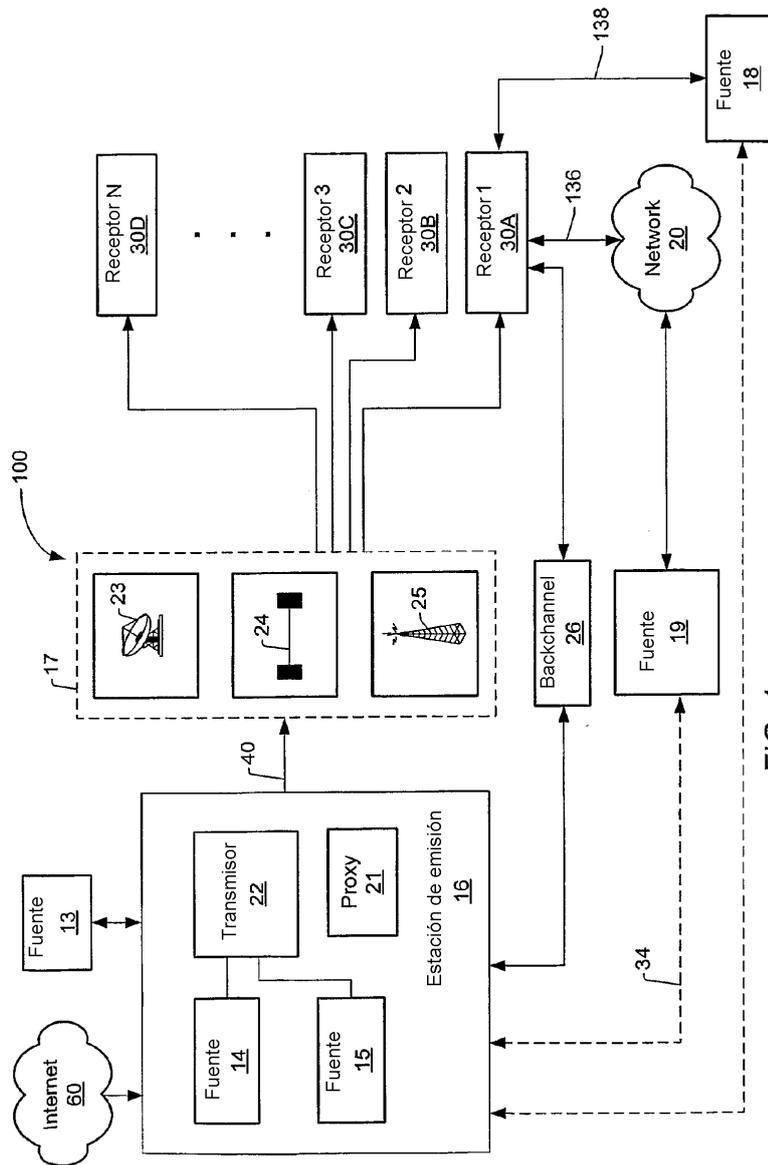


FIG. 1

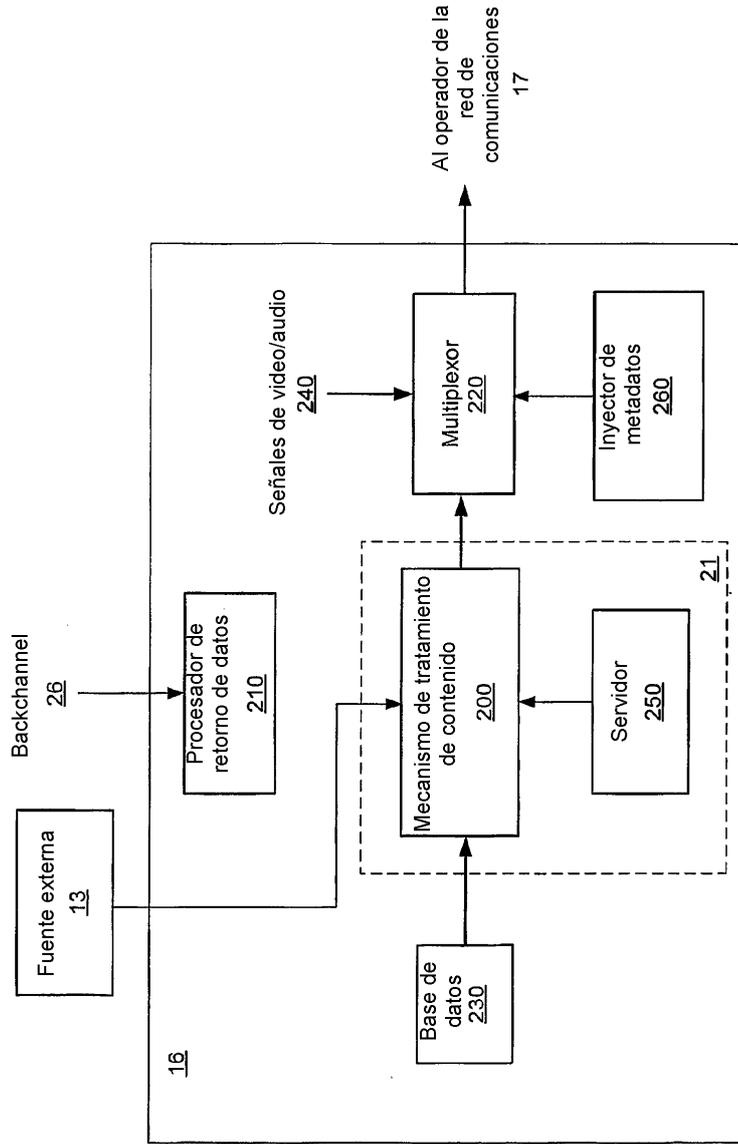


FIG. 2

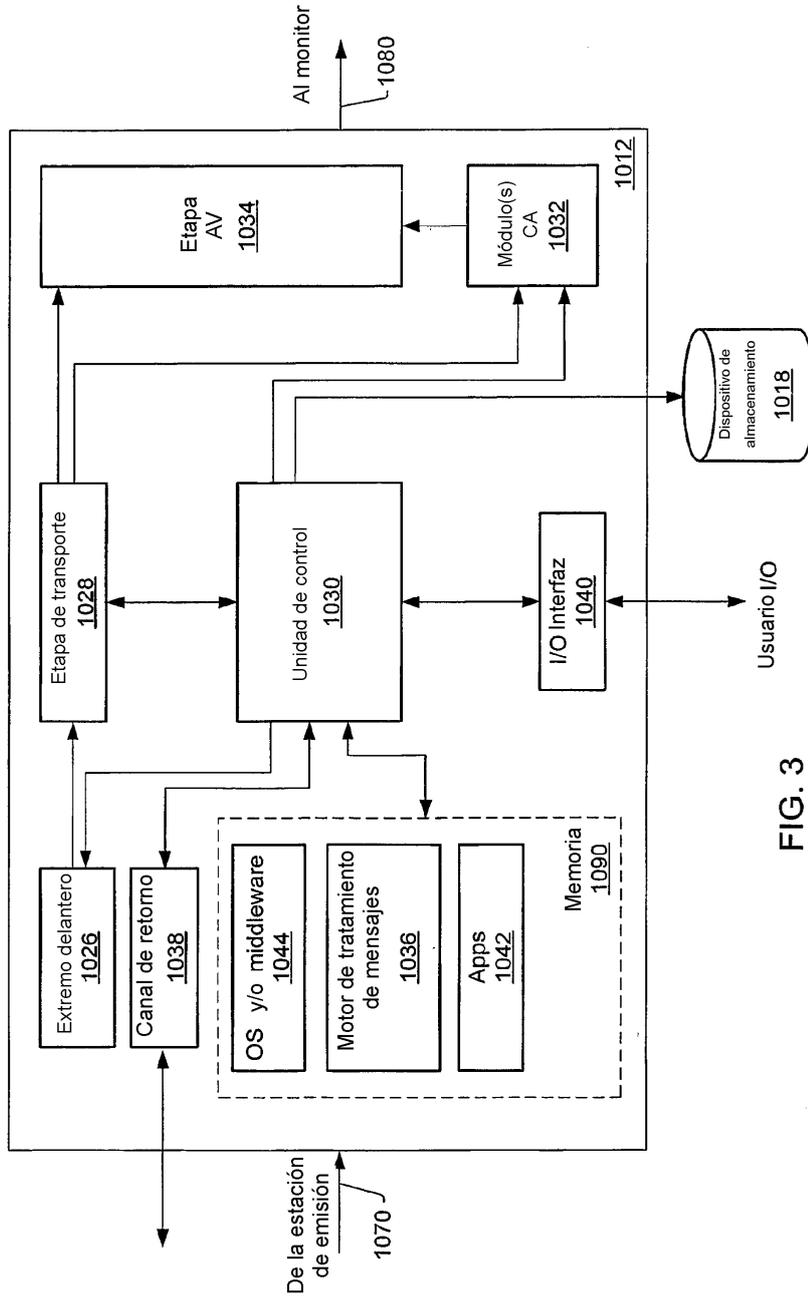


FIG. 3

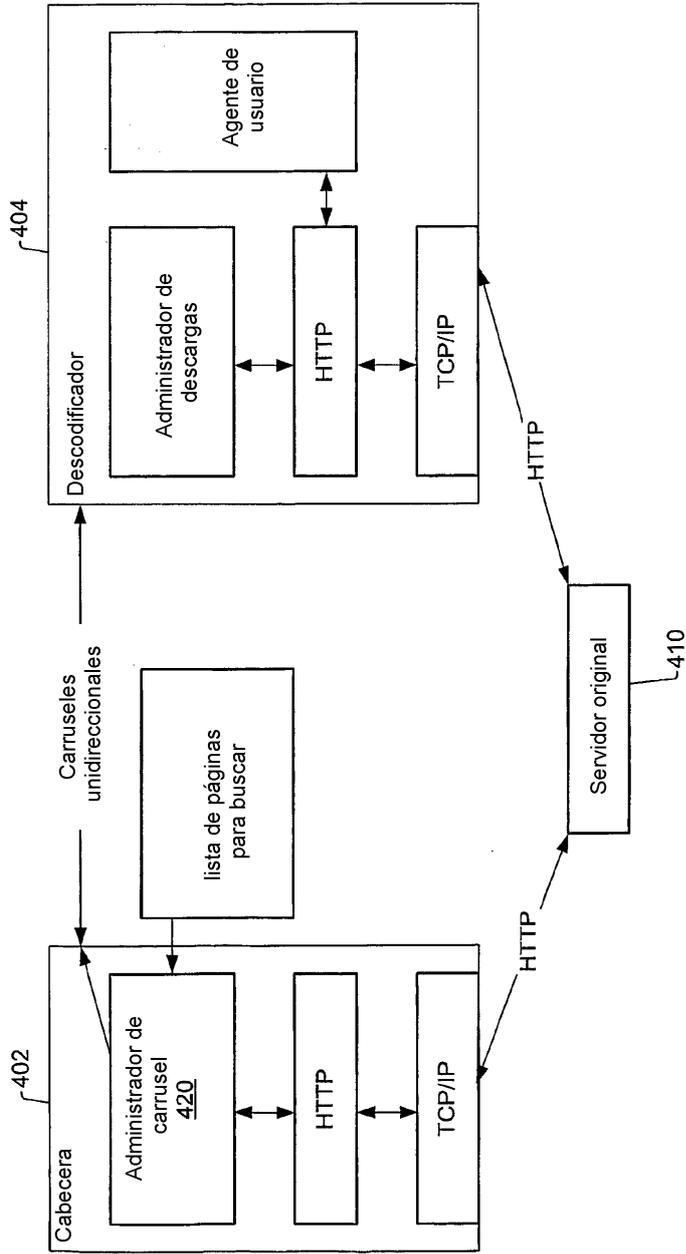


FIG. 4

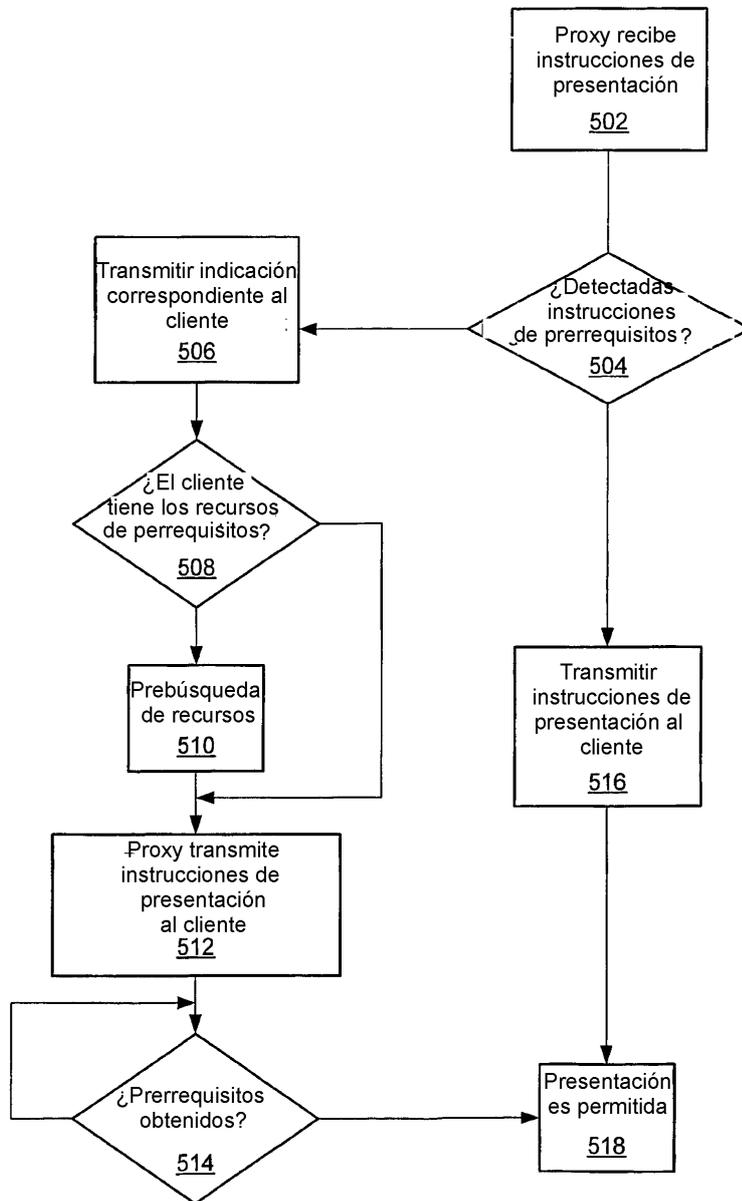


FIG. 5