

Actualización de datos desde carrusel en aplicaciones GINGA

Informe de Proyecto
ELO323 Redes de Computadores II

Integrantes:

Tania Espinoza

Andrés Gómez

Profesor:

Agustín González

Fecha:

15 de enero de 2014

Tabla de Contenidos

Introducción	3
DSM-CC (Digital Storage Media – Command and Control)	3
Introducción	3
Carrusel de Datos	3
Carrusel de objetos	4
Eventos de flujo.....	4
Transporte de comandos de edición NCL.....	5
Introducción	5
Transporte de comandos por descriptores de eventos de flujo y carrusel de objetos DSM-CC	5
Transporte de Estructuras de metadatos como parámetro de comando de edición	7
Transporte de Estructuras de metadatos en secciones de metadatos MPEG-2	7
Comandos de edición NCL.....	8
Introducción	8
Comandos de edición NCL.....	8
Resultados.....	10

Introducción

Este informe tiene como finalidad presentar la información necesaria que permita comprender el mecanismo de transmisión de comandos de edición para su uso en aplicaciones NCL. La principal fuente de información se adquirió del documento “Programando em NCL” de los autores Luis Fernando Gomes Soares y Simone Diniz Junqueira Barbosa, publicado por la Pontificia Universidade do Rio de Janeiro, que es la principal fuente de información disponible sobre este tema, siendo referido en la mayoría de los sitios web dedicados al uso de estos comandos.

DSM-CC (Digital Storage Media – Command and Control)

Esta sección del informe pretende presentar una visión general del protocolo DSM-CC y analizar los casos que permitan el entendimiento de la forma en que el protocolo actúa en la difusión de datos en el sistema de televisión digital.

Introducción

El protocolo DSM-CC fue originalmente ideado para su implementación mediante algún tipo de mecanismo RPC (Remote Procedure Call), es decir, donde los datos serían buscados (pushed from) en un proveedor de contenidos. Los sistemas de difusión como la TV digital terrestre, tienen una naturaleza distinta ya que los datos son enviados (pushed to) desde el proveedor para los clientes sin haber solicitud alguna. Por esto, fue necesaria la búsqueda de una solución para el acceso a los datos.

La solución encontrada es bastante simple, sin embargo menos eficiente: los sistemas de archivos deben ser periódicamente transmitidos por el servidor de contenidos, debiendo el cliente receptor esperar por el archivo que desea. Este tipo de solución es llamada carrusel.

En DSM-CC se da soporte a dos tipos de carrusel: carrusel de datos y carrusel de objetos.

Carrusel de Datos

Un carrusel de datos es la forma más simple de transmisión de datos DSM-CC. No existe especificación sobre la información que deben proveedor los datos, por lo que es tarea del receptor analizar los datos de modo de que adquieran un sentido para él.

Un carrusel de datos consiste en un número de módulos, en que cada módulo contiene algún ítem de datos, como por ejemplo un archivo. No existe ninguna estructuración de más alto nivel por encima del módulo.

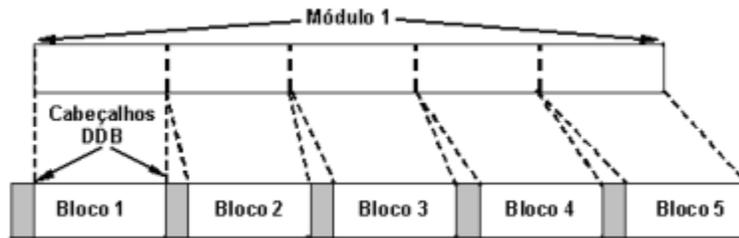


Fig 1: Módulo de un carrusel

Carrusel de objetos

Los carruseles de objetos son construidos teniendo como base el modelo de carrusel de datos, agregándoles los conceptos de archivos, directorios y flujos.

Los datos son representados por objetos (objetos de archivos, objetos de directorio, objeto de flujo, objeto de evento y objeto Service Gateway, lo cuales contienen atributos (nombre, tipo, y posiblemente, contenido).

Los objetos de flujo permiten a un carrusel de objeto referenciar flujos elementales que son parte de la difusión. Para hacer un único tap (referencia), que se refiera a un flujo completo, o se pueden hacer varios taps, que hagan referencia a varios flujos que forman un único flujo lógico. Por ejemplo, un programa de TV puede contener dos pistas de audio correspondientes a idiomas diferentes.

Eventos de flujo

Los eventos de flujos son descriptores insertados dentro de un flujo elemental DSM-CC. Estos descriptores permiten la sincronización de eventos con un flujo de media. Tal como los objetos de flujo, los eventos de flujo contienen una lista de taps que hacen referencia a flujos elementales.

Los eventos de flujo con bastantes útiles para especificar eventos no previsibles. Por ejemplo, en un partido de futbol, al momento de un gol se puede querer sincronizar con otro objeto de media cualquiera, como por ejemplo un ranking de los delanteros del campeonato.

Desde el punto de vista de la especificación DSM-CC, los tratamientos de eventos de flujo son divididos en dos partes:

1. Objetos de eventos de flujo: transportados en el carrusel DSM-CC.
2. Descriptores de eventos de flujo: transportados en secciones privadas DSM-CC.

Un descriptor de evento de flujo determina el disparo de un evento que puede ser referido por medio de un objeto de eventos de flujo, que describe en más alto nivel qué es lo que significa el

evento. Más de un descriptor de evento de flujo puede ser referido mediante un mismo objeto de evento de flujo.

Un objeto de flujo puede ser un identificador (eventId) que debe ser único dentro de un carrusel, y un nombre legible para un ser humano, como por ejemplo “nclEditingCommand”. Una aplicación se puede registrar para recibir eventos con ese nombre legible. Por ejemplo, el Administrador de Base Privada del ambiente declarativo Ginga-NCL se registra para recibir eventos nclEditingCommand que corresponden a comandos de edición de documentos NCL.

Los descriptores de evento son transportados en flujos listados en una tabla PMT, y poseen además de los identificadores pertinentes, una referencia temporal la cual indica al receptor el instante en que el evento deberán ejecutarse, lo cual usualmente está basado en un flujo denominado Normal Play Time (NPT).

Transporte de comandos de edición NCL

Tal como se ha señalado en la primera sección de este informe, los comandos de edición NCL pueden ser enviados de diversos modos. En esta sección detallan los mecanismos de transporte de estos comandos, que puede ser llevado a cabo mediante estructuras de datos recibidas sin solicitud como a través de estructuras de datos recibidas por demanda, pero especialmente utilizando secciones MPEG-2.

Introducción

Los parámetros de los comandos de edición pueden ser insertados en un descriptor de evento de comando o pueden ser enviados estructuras de datos externas, referenciadas por los descriptores de evento, siendo lo segundo siempre el caso de los comandos de edición addDocument y addNode.

Transporte de comandos por descriptores de eventos de flujo y carrusel de objetos DSM-CC

En los sistemas de TV digital terrestre es común el uso del protocolo DSM-CC para el transporte de comandos de edición en flujos elementales de transporte MPEG-2.

DSM-CC (en inglés “Digital Storage Media Command and Control”), consiste en un marco de desarrollo para controlar los canales asociados a los flujos de datos MPEG-1 y MPEG-2; y utiliza un modelo cliente-servidor sobre una red subyacente. El uso de DSM-CC está generalmente relacionado a la distribución de video (vía satélite o vía terrestre) con contenido interactivo; pero también tiene aplicaciones entre servidores y clientes de audio.

Los comandos de edición NCL pueden ser transportados usando descriptores de evento de flujo DSM-CC. Los descriptores de evento de flujo DSM-CC tienen una estructura muy parecida a los descriptores de evento NCL presentados en la sección anterior de este informe.

Sintaxe	N.º de Bits
StreamEventDescriptor () {	
<i>descriptorTag</i>	8
<i>descriptorLenght</i>	8
<i>eventId</i>	16
<i>reserved</i>	31
<i>eventNPT</i>	33
<i>privateDataLength</i>	8
<i>commandTag</i>	8
<i>sequenceNumber</i>	7
<i>finalFlag</i>	1
<i>privateDataPayload</i>	8 a 2008
<i>FCS</i>	8
}	

Tabla 1: Descriptor de evento de flujo para comandos de edición

Para la transmisión de parámetros de comandos de edición, el carrusel de objetos DSM-CC puede ser utilizado. El protocolo de carrusel de objetos DSM-CC permite una transmisión cíclica de objetos de eventos de flujo y sistemas de archivos.

En un carrusel de objetos, los objetos de evento de flujo son utilizados para mapear los nombres de los eventos de flujo en id's de eventos de flujo, definidos para los descriptores de evento. En el caso de evento de comando NCL, sería responsable de mapear el nombre `nclEditingCommand` en `eventIds`, definido en los descriptores de eventos. Los nombres de los eventos permiten especificar tipos de eventos, ofreciendo un mayor nivel de abstracción a las aplicaciones. El administrador de base privada, debe registrar como observador los eventos con los que se enlaza, utilizando nombres de evento; en el caso de comandos de edición, el nombre es "nclEditingCommand".

Además de los objetos de eventos de flujo, el carrusel de objetos DSM-CC puede también ser utilizado para el transporte de archivos organizados en un directorio. Un demultiplexador DSM-CC es responsable de montar la estructura recibida en el dispositivo receptor. Los parámetros de comandos de edición especificados como documentos XML (documentos NCL o entidades NCL que se quieran adicionar) pueden de este modo ser organizados en un sistemas de archivos a ser transportados por el carrusel, esto como una alternativa de transporte dentro de los propios descriptores de eventos que definen los comandos. El generador de carrusel de objetos es el responsable de juntar estos archivos y los objetos de eventos de flujo en los flujos elementales de MPEG-2.

De este modo, cuando un comando de edición NCL requiere ser enviado, un objeto de evento DSM-CC debe ser creado, mapeando un string "nclEditingCommand" en una id de flujo de evento,

y luego colocado en un carrusel de objetos DSM-CC, que es enviado en un flujo elemental. Uno o más descriptores de evento de flujo DSM-CC con la id previamente seleccionada pueden ser entonces creados y enviados en otro flujo elemental MPEG-2 TS. Estos eventos de flujo pueden tener una referencia de tiempo colocada en cero, o puede ser definida para su ejecución en un tiempo específico. El administrador de base privada debe encargarse de interpretar el string “nclEditingCommand” y será notificado cuando esos flujos de eventos lleguen. El commandTag recibido es entonces utilizado por el administrador de base privada para interpretar la semántica del command string.

En el caso de un descriptor de evento de comando de edición, si el “command parameter” basado en XML es lo suficientemente corto, éste puede ser transportado directamente en la carga útil (payload) de los descriptores de eventos. En caso contrario, privateDataPayload transportará un conjunto de pares de referencia. En este caso, la especificación XML debe ser enviada en el mismo carrusel que transporta los objetos de eventos. El parámetro uri del primer par de referencia debe tener el camino absoluto a la especificación XML (la ruta en el servidor de datos). El parámetro id correspondiente al par de referencia, permite distinguir la id del carrusel, del módulo pertinente del carrusel y el “objectKey” en el carrusel de datos. En comandos de edición distintos a addDocument y addNode, el parámetro uri puede ser omitido. Si otros sistemas de archivos precisan ser transmitidos usando otros carruseles de datos a fin de completar el comando de edición (lo que es usual en los comandos addDocument y addNode) con contenido de media, otros pares {uri,id} deben estar presentes en el comando. En este caso, el parámetro uri debe tener el camino absoluto a la raíz del sistema de archivos, y el parámetro id debe permitir la identificación de cualquier archivo en el directorio raíz en el carrusel de objetos.

Transporte de Estructuras de metadatos como parámetro de comando de edición

Una alternativa de transporte de estructuras de metadatos directamente en secciones NCL consiste en tratar estas estructuras como parámetros de los comandos addDocument y addNode, transportados en el campo privateDataPayload de los descriptores de eventos.

En este caso, el conjunto de pares {uri, id} de los comandos addDocument y addNode son sustituidos por parámetros de la estructura de metadatos, que definen un conjunto de pares {“uri”, “component_tag, structureId”} para cada archivo transmitido sin previa solicitud.

Transporte de Estructuras de metadatos en secciones de metadatos MPEG-2

Otra alternativa consiste en transportar las estructuras de metadatos en secciones de metadatos MPEG-2, transportadas en flujos MPEG-2. Cada sección de metadatos MPEG-2 podrá contener

datos de sólo una estructura de datos. Sin embargo, una estructura de metadatos se puede extender en varias secciones de metadatos.

Comandos de edición NCL

Los comandos de edición NCL pueden llegar al receptor de diversas maneras: mediante la red (canal de difusión o canal de interactividad en cualquier otra red, tal como Internet), a través de objetos imperativos insertos en un propio documento NCL, o mediante alguna aplicación externa controlada por un usuario.

Mediante los comandos de edición, se pueden crear y modificar documentos NCL en el transcurso del tiempo de exhibición.

Introducción

El mecanismo utilizado para la presentación de documentos NCL en el dispositivo receptor consiste principalmente en un formateador NCL y un módulo administrados de base privada, siendo el primero el encargado de recibir un documento NCL y controlar su presentación, debiendo garantizar que las relaciones especificadas entre los objetos sean respetadas. El formateador debe trabajar con aplicaciones NCL que son recolectadas dentro de una estructura de datos conocida como base privada. A modo de ejemplo, en el estándar de televisión IDSB-Tb, el middleware Ginga asocia por lo menos una base privada a un canal de televisión.

Los documentos en una base privada pueden ser inicializados, pausados, reanudados, detenidos y referidos unos a otros.

El administrador de base privada es el responsable de recibir los comandos de edición de documentos NCL y ejecutarlos, lo cual puede provocar una modificación en los documentos NCL activos, es decir siendo presentados. Por tanto se incorpora la capacidad de edición en vivo de las aplicaciones.

Estos comandos de edición pueden ser recibidos por distintas vías. Pueden ser transmitidos en un ambiente de TV digital terrestre mediante el canal de difusión, para lo cual es usual adoptar el protocolo DSM-CC para el transporte de comandos de edición generados por los proveedores de las emisoras de televisión. También es posible recibir comandos de edición mediante el canal de interactividad.

Comandos de edición NCL

Los comandos de edición NCL son desarrollados en una estructura llamada descriptor de evento. Cada descriptor de evento (de edición) tiene una estructura compuesta básicamente por un id, una referencia de tiempo y un campo de datos privados. Un identificador define de forma exclusiva cada evento de edición.

La referencia de tiempo indica el momento exacto en que se debe ejecutar el evento. Una referencia de tiempo igual a cero indica que el evento de edición debe iniciarse inmediatamente

después de ser recibido, siendo los eventos de este tipo comúnmente conocidos como eventos “do it now”.

El campo de datos privados ofrece soporte para la identificación del comando de edición y la definición de los parámetros del evento de edición, tal como se presentan en la tabla 1.

Sintaxe	Número de Bits
EventDescriptor () {	
eventid	16
eventNPT	33
privateDataLength	8
commandTag	8
sequenceNumber	7
finalFlag	1
privateDataPayload	8 a 1928
FCS	8
}	

Tabla 2: Descriptores de eventos para comando de edición

El campo commandTag identifica el tipo de comando de edición de manera exclusiva. Para permitir el envío de comando completos con más de 255 bytes en más de un descriptor de evento, todos los descriptores de un mismo comando deben ser numerados y enviados en secuencia, de modo de que puedan ser multiplexados con otros comando de edición con el mismo commandTag, siendo en este caso finalFlag=1 para todos los descriptores de evento, con excepción del descriptor final que tiene finalFlag=0. El campo privateDataPayload contiene los parámetros del comando de edición. Finalmente, el campoFCS contiene un checksum de todo el campo privateData.

Los comandos de edición NCL son divididos en tres grupos:

- Comandos de edición de base privada: para abrir, activar, desactivar, cerrar y guardar bases privadas.
- Comandos de manipulación de documentos: para agregar, remover y guardar documentos en una base privada abierta, y para iniciar, pausar, reanudar y parar presentaciones de documentos en una base privada activa.
- Comando para la manipulación de entidades NCL de un documento en una base privada abierta. Para cada entidad NCL, fueron definidos los comandos add y remove.

El comando add siempre tiene entidades NCL como argumento (parámetros de comando especificados en XML). Las entidades son definidas utilizando sintaxis idéntica a la utilizada en los esquemas NCL, con la excepción del comando addInterface: el atributo para comenzar un

elemento <area> puede recibir un valor "now", especificando un valor NPT (Normal Play Time) actual de un objeto determinado por un argumento nodeId. La consistencia del documento es mantenida por el formateador NCL, que debe asegurar que todos los atributos de identidad obligatorios sean definidos.

Si el parámetro de comando basado en XML es lo suficientemente breve, éste puede ser transportado en la carga útil (payload) del descriptor de evento. En caso contrario, privateDataPayload transporta un conjunto de pares de referencia {uri,id}, con una interpretación que se señala a continuación.

En el caso de archivos recibidos por un canal de difusión (documentos NCL u objetos de documentos NCL enviados sin ser solicitados), cada par relaciona la dirección del archivo en el directorio y su respectiva localización en el sistema de transporte (carrusel de datos). No es necesario la incorporación de un par {uri,id} en el comando para cada archivo enviado por difusión. Pero es necesario que desde el par {uri,id} incluido en el comando, todo archivo recibido posea su vía de acceso única (uri).

En el caso de archivos recibidos por demanda mediante un canal de interactividad o localizados en el propio receptor, ninguna de estas referencias necesita ser enviadas, a menos de que especifique un documento NCL o una especificación XML a un objeto NCL que deberá ser agregado, en este caso se debe enviar un par {uri,"null"}, especificado la ruta en que el archivo debe ser buscado.

Resultados

La información recopilada en este informe ha permitido obtener una base teórica que permite la comprensión del mecanismo de transmisión de comandos de edición para aplicaciones GINGA. Del mismo modo, se ha logrado una comprensión de las posibilidades que ofrece el uso de esta tecnología, permitiendo crear aplicaciones interactivas editadas en tiempo real.

La presentación de un ejemplo demostrativo se ve dificultada debido a que no se disponen de televisores o set-top-boxes que estén implementados para el uso de comandos de edición en aplicaciones GINGA, por lo que no es posible mostrar los cambios de una aplicación de manera sencilla.

Sin embargo, se dispone de aplicaciones que han sido usadas para ser modificados mediante comandos de edición, lo que permite visualizar una forma de implementación de esta tecnología en una aplicación final. Estos en conjunto, con los conceptos teóricos presentados permitirían tener una visión del mecanismo de comando de edición, sin embargo una demostración práctica es un desafío vigente.