

## Mecanismo de Identificação de Recursos para Aplicações Interativas em Redes de TV Digital por Difusão

Marcio Ferreira Moreno, Rogério Ferreira Rodrigues, Luiz Fernando Gomes Soares

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

Caixa Postal 38.097 – 22.453-900 – Rio de Janeiro – RJ – Brazil

{mfmoreno, rogerio, lfgs}@inf.puc-rio.br

**Abstract.** *This paper describes a resource identifier mechanism for interactive applications that are found in broadcast digital TV networks. The considered mechanism is based on MPEG-2 System and DSM-CC standards, and relieves application authors to have knowledge of the addressing scheme used to identify resources in DTV presentation environments (set-top boxes or other receivers) and transport systems. The proposal comes from the reference middleware defined for the Terrestrial DTV Brazilian System, but can be applied to other networks and DTV systems.*

**Resumo.** *Este artigo descreve um mecanismo de identificação de recursos para aplicações interativas em redes de TV digital por difusão. O sistema proposto, construído tendo por base os padrões MPEG-2 System e DSM-CC, desobriga os autores das aplicações de conhecer as idiossincrasias da identificação de recursos nos ambientes de exibição (receptores nos ambientes dos telespectadores) e nos sistemas de transporte. A proposta é parte do modelo de referência do middleware do sistema brasileiro de TV digital terrestre, mas pode ser aplicada em outras redes e outros sistemas de TV digital.*

### 1. Introdução

Na maioria dos sistemas de TV digital terrestre existentes [Morris and Chaigneau 2005], o formato utilizado para multiplexação de dados (incluindo as aplicações interativas) e fluxos de áudio e vídeo é especificado pelo padrão MPEG-2 Sistemas [ISO 2000], mais especificamente, o fluxo de transporte MPEG-2 (MPEG-2 *Transport Stream*, ou MPEG-2 TS). A sintonização de um canal específico de TV (e, assim, um fluxo TS específico), no entanto, pode ser realizada em qualquer instante, obrigando que dados que não tenham relação temporal por *timeline* (mais especificamente por selos de tempo) sejam enviados ciclicamente em um fluxo TS, para que seus recebimentos sejam independentes do instante de sintonização. Para dar suporte a esse envio cíclico, prevalece nos sistemas de TV digital terrestre a utilização do protocolo carrossel de objetos, especificado no padrão DSM-CC (*Digital Storage Media – Command and Control*) [ISO 1998]. O Sistema Brasileiro de TV Digital Terrestre, denominado ISDTV-T (*International Standard for Digital TV - Terrestrial*), também optou pela

utilização desses padrões para darem suporte à transmissão e multiplexação dos fluxos de áudio, vídeo e dados.

Uma das principais características do carrossel de objetos é permitir o envio cíclico de um sistema de arquivos, do provedor de conteúdo (ambiente de autoria) para os receptores (ambiente de exibição). Assim, ao sintonizar um determinado canal, o receptor deve ser capaz de decodificar os dados recebidos e colocá-los em uma área de memória para que possam ser utilizados, preservando a estrutura de arquivos e diretórios enviada. Cabe observar que mais de um carrossel pode ser transmitido simultaneamente, e que um carrossel de objetos pode fazer uso de recursos (arquivos, diretórios e outros objetos [ISO 1998]) que estejam sendo transferidos em outros carrosséis. Mais ainda, aplicações transferidas em arquivos de um carrossel podem fazer referência a recursos presentes no mesmo carrossel ou a recursos de outros carrosséis. Por exemplo, uma página HTML transmitida em um carrossel pode referenciar uma imagem cujo conteúdo é transmitido em um outro carrossel de objetos.

Na aplicação cliente (receptor de TV) – servidor (provedor de conteúdo, por exemplo, estação radiodifusora) definida por um sistema de TV digital terrestre, uma questão importante é a definição de um mecanismo para a identificação e localização dos recursos no ambiente do cliente, no ambiente do servidor e nas estruturas transmitidas pelo carrossel de objetos. Porém, é desejável que tal mecanismo livre os autores das aplicações, que operam no lado servidor, de conhecer como as estruturas de dados serão transmitidas e armazenadas no cliente, onde as aplicações serão exibidas.

Inicialmente, na autoria, os recursos (arquivos de vídeo, de imagem, de texto, de áudio, de scripts etc.) utilizados para compor os programas de TV interativos estão localizados na máquina em que é feita a autoria, ou em sistemas de armazenamento (provedores de conteúdo) distribuídos em uma rede à qual o ambiente de autoria encontra-se conectado. Para identificar cada um dos recursos, geralmente são utilizadas URLs (*Universal Resource Locators*), cujos esquemas [RFC 2005] identificam os recursos com base em suas localizações. Ao serem recebidos pelo receptor, os recursos precisam ter suas identificações atualizadas, já que muitos deles possuirão uma nova localização. Considerando também o fato de que um carrossel de objetos, e aplicações transmitidas neste carrossel, podem referenciar recursos transmitidos em outros carrosséis, é preciso que haja uma forma de traduzir a identificação dos recursos no ambiente de autoria em uma identificação dos recursos para o ambiente de transmissão, e deste, para o ambiente de exibição. Uma questão importante nessa tradução é o fato de que embora o carrossel de objetos preserve a estrutura de arquivos e diretórios enviada, sem informações adicionais, o receptor não sabe sob que diretório pai deve colocar a raiz do diretório recebido no carrossel.

Nos provedores de conteúdo para TV digital terrestre existentes, a tradução é feita no ambiente de autoria, automaticamente ou pelo próprio autor. No entanto, essa estratégia sobrecarrega o ambiente de autoria, impõe ao autor o conhecimento do sistema de transporte e pode introduzir erros de especificação ou atrasos na geração e transmissão dos programas. Este artigo propõe um mecanismo de identificação única, tanto no ambiente de autoria como no ambiente de exibição, tornando desnecessário para o autor de aplicações interativas o conhecimento das idiossincrasias da identificação e localização de recursos de cada um dos ambientes. A proposta é parte da

especificação de codificação de dados do Sistema Brasileiro de TV Digital Terrestre e da implementação de referência do middleware Ginga em receptores para tal sistema.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os conceitos e as tecnologias envolvidos no sistema proposto. A Seção 3 realiza um breve resumo dos trabalhos relacionados na área. A Seção 4 descreve o sistema proposto e cita as implementações realizadas. Finalmente, a Seção 5 é reservada para as considerações finais.

## 2. Conceitos e Tecnologias Relacionados

Como acontece com qualquer sistema distribuído, em um sistema de TV digital, os equipamentos dos provedores de conteúdo e os equipamentos receptores devem compartilhar de protocolos bem definidos para que o intercâmbio de informações e a provisão de serviços sejam possíveis. As seções a seguir detalham padrões e protocolos, alguns já introduzidos na Seção 1, utilizados com esse propósito. Esta seção se faz necessária para o entendimento dos trabalhos relacionados e da solução proposta neste artigo.

### 2.1. Fluxo de Transporte MPEG-2

As especificações do fluxo de transporte fazem parte do padrão MPEG-2 *Systems* [ISO 2000], ou MPEG-2 *Sistemas*, e estabelece como um ou mais sinais de áudio e vídeo, assim como outros dados (imagens estáticas, textos etc.), devem ser combinados de forma a serem transmitidos. Um fluxo de transporte é formado por um ou mais fluxos elementares (ES – *Elementary Stream*). Um *fluxo elementar* é definido como um fluxo de dados gerado pela codificação do conteúdo de vídeo, áudio ou outros dados.

As especificações MPEG-2 *Sistemas* definem ainda o termo *programa*, chamado de *serviço* no contexto da TV digital, como um grupo composto de um ou mais fluxos elementares com uma mesma base temporal [ISO 2000]. O fluxo de transporte pode conter vários serviços (programas) simultaneamente, cada um podendo ter uma base de tempo diferente.

Simplificadamente, multiplexar serviços em um fluxo de transporte significa organizar os pacotes dos vários fluxos elementares, pertencentes aos serviços contemplados, em um único fluxo. Para isso, é necessário inserir no fluxo de transporte informações que permitam ao decodificador MPEG-2 identificar a qual serviço um dado fluxo elementar pertence. Essas informações são dispostas como um conjunto de tabelas de informação específica de programa (PSI – *Program Specific Information*). Uma PSI particular, denominada PMT (*Program Map Table*), contém a lista de identificadores dos fluxos elementares que compõem um serviço. Cada PMT encontrada representa um serviço disponível. As PMTs são localizadas através de outra PSI denominada PAT (*Program Association Table*), que contém identificadores dos fluxos elementares contendo as PMTs. O fluxo elementar que possui a PAT possui identificador fixo com o valor hexadecimal 0x00.

### 2.2. Eventos de Sincronismo DSM-CC

Sincronizar o comportamento de uma aplicação com o conteúdo de uma programação de TV específica (áudio e vídeo principal) é extremamente desejável, principalmente

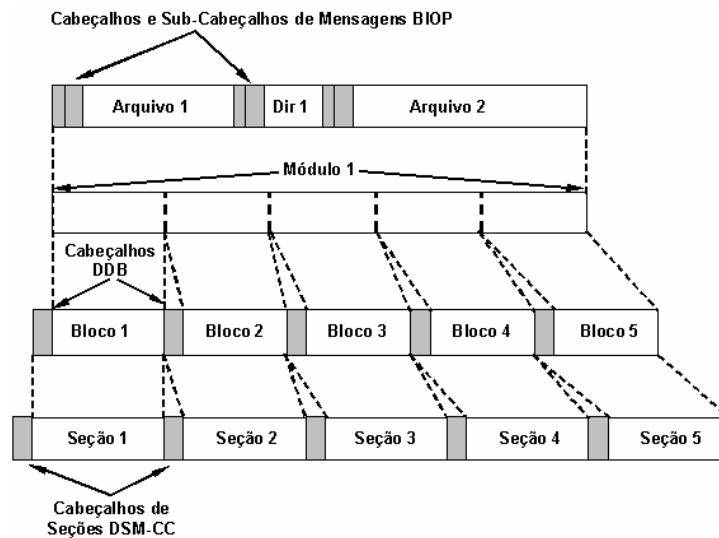
quando a aplicação em questão possui relação semântica com essa programação. O sincronismo pode ser realizado através de *selos de tempos* (paradigma *timeline*), quando os dados das aplicações são enviados de forma síncrona, ou sincronizada [ISO 2000] com o áudio e vídeo principal, ou alternativamente através de *eventos de sincronismo DSM-CC* (ou simplesmente eventos DSM-CC), quando os dados são transmitidos de forma assíncrona (sem o uso de selos de tempo, utilizando, por exemplo, o carrossel de objetos, a ser discutido na Seção 2.3).

Para criar um evento DSM-CC, é inserida uma estrutura no fluxo de transporte, denominada descritor de eventos. Cada *descritor de eventos* possui um identificador numérico único, que o identifica no fluxo de transporte. Uma vez que não existe a possibilidade do provedor de conteúdo saber exatamente a posição que o descritor será inserido no fluxo, cada descritor possui uma referência temporal que indica em qual instante o evento deverá ocorrer, usualmente baseado em um fluxo DSM-CC denominado NPT (*Normal Play Time*) [ISO 1998]. Como caso particular, um descritor de eventos pode informar ao sistema receptor que o evento deve ocorrer imediatamente – esse tipo de evento é chamado de evento “*do it now*”. Além do identificador e da referência temporal, o descritor de eventos possui também um campo para dados específicos das aplicações, que pode ser utilizado de acordo com sintaxes e semânticas a serem tratadas pelas próprias aplicações. Para evitar que um descritor de evento DSM-CC seja perdido devido a falhas no meio de difusão, um mesmo descritor normalmente é enviado diversas vezes pelo provedor de conteúdo, cabendo ao receptor interpretar esses eventos uma única vez.

Os descritores de eventos DSM-CC, bem como os outros dados DSM-CC (por exemplo, dados referenciados pelos eventos e que devem ser sincronizados com o áudio e vídeo principal), precisam de estruturas especiais para serem transportados. Para tanto, as especificações do protocolo DSM-CC determinam estruturas de dados denominadas *seções DSM-CC*. Essas estruturas possuem um cabeçalho, especificado com o objetivo de informar ao decodificador como as seções estão sendo utilizadas para transportar os dados, como elas devem ser remontadas, quais os tipos de dados transportados, além de outros parâmetros para o tratamento apropriado da informação [ISO 1998].

### 2.3. Carrossel de Objetos DSM-CC

Conforme mencionado na Seção 1, *carrossel de objetos* é um protocolo de transmissão cíclica de dados. Os dados são representados por *objetos* (objeto de diretório, objeto de arquivo etc. [ISO 1998]), que contêm atributos (nome, tipo e, possivelmente, conteúdo). Segundo as especificações DSM-CC, que são compatíveis com o framework ORB (*Object Request Broker*) definido pelas especificações CORBA (*Common Object Request Broker Architecture*) [OMG Specification 2004], cada objeto deve ser encapsulado em uma mensagem BIOP (*Broadcast Inter ORB Protocol*), que são transmitidas em módulos. Uma mensagem BIOP deve ser transmitida em um único módulo, mas um módulo pode conter mais de uma mensagem, conforme ilustra a Figura 1. Os módulos, por sua vez, são divididos em blocos de dados, que são encapsulados em mensagens denominadas DDB ou *DownloadDataBlock* [ISO 1998]. Os blocos de dados, por sua vez, são encapsulados em seções DSM-CC. As regras de encapsulamento de blocos de dados em seções foram especificadas para que blocos possam ser adquiridos diretamente do fluxo de transporte através de filtros [ETSI Standard 2003b].

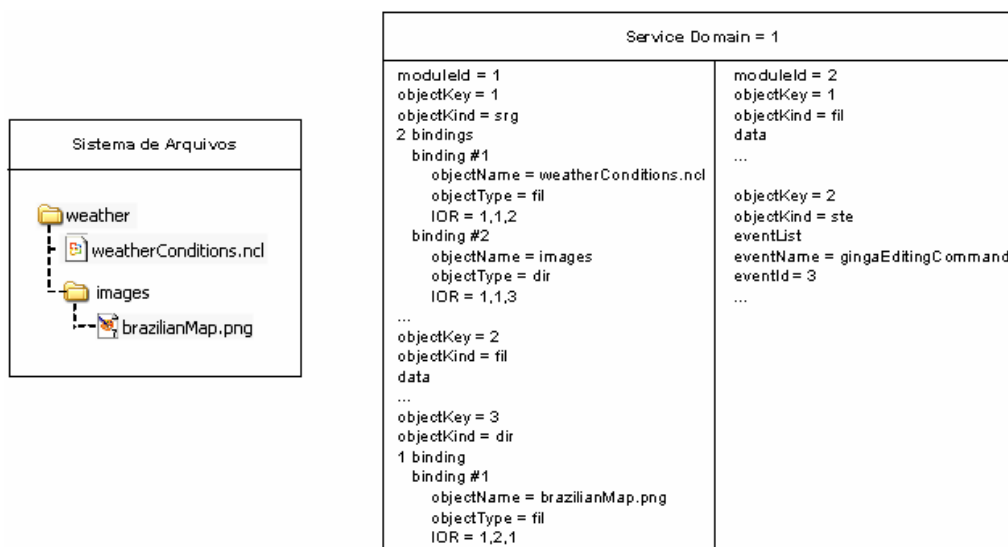


**Figura 1. Processo de Encapsulamento do Carrossel de Objetos DSM-CC**

Cada seção DSM-CC de um carrossel é transmitida no fluxo de transporte, como um fluxo elementar de dados, uma após a outra. Depois de transmitir a última seção (no exemplo da Figura 1, a “Seção 5”), a transmissão é reiniciada. O resultado disso é um fluxo elementar que contém o sistema de arquivos transmitido de forma cíclica. Assim, se um determinado terminal de acesso não recebeu um bloco de dados em particular (devido a uma falha na transmissão ou por ter sintonizado o canal após a transmissão desse bloco), basta esperar pela retransmissão correta da seção contendo esse bloco de dados.

Cada instância de carrossel de objetos é representada por um *Service Domain* [ISO 1998], que consiste em uma identificação única do carrossel de objetos. Todo *Service Domain* possui um *Service Gateway* [ISO 1998], que contém referências para todos os objetos dispostos na raiz do carrossel de objetos. O padrão DSM-CC utiliza a estrutura de referências IOR (*Interoperable Object Reference*), definidas nas especificações CORBA. No contexto do protocolo carrossel de objetos, uma IOR é normalmente composta pelo identificador do carrossel, seguido do identificador do módulo e pelo identificador do objeto. O *Service Gateway* é um objeto do carrossel de objetos cuja localização é transmitida em uma mensagem *DownloadServerInitiate* (DSI) [ISO 1998], que pode ser encontrada através das informações PSI (discutidas na Seção 2.1).

Como exemplo, considere o sistema de arquivos apresentado na Figura 2. Considere ainda um carrossel de objetos gerado a partir desse sistema de arquivos, com o *Service Domain* de valor hexadecimal “1”, conforme apresentado também na Figura 2.



**Figura 2. Exemplo de Carrossel de Objetos gerado a partir de um sistema de arquivos**

Nesse *Service Domain*, dois módulos são gerados e identificados com valores em hexadecimal “1” e “2”. O identificador de cada objeto é apresentado através do campo “objectKey”. O objeto que representa o *Service Gateway* (na Figura 2, objeto do tipo “srg”) é identificado com o valor “1” e é encapsulado no módulo “1”. Como consequência, a IOR do objeto *Service Gateway* transmitida na mensagem *DownloadServerInitiate* é definida por “1,1,1” (*Service Domain* = 1, id do módulo = 1 e id do objeto = 1). Os objetos que representam o arquivo “weatherConditions.ncl” e o diretório “images” são identificados pelos valores hexadecimais “2” e “3”, respectivamente, e também são encapsulados no módulo “1”. Já o objeto que representa o arquivo contendo o conteúdo da imagem é identificado com o valor “1”, mas encapsulado no módulo “2”. O objeto *Service Gateway* possui, por sua vez, duas IORs. Para relacionar uma IOR ao nome do objeto que a mesma referencia, bem como ao tipo desse objeto (arquivo, diretório etc.), as especificações DSM-CC utilizam o conceito de *binding*. Assim, no exemplo, o objeto *Service Gateway* possui dois *bindings* para os dois objetos do módulo “1” que são filhos diretos da raiz do sistema de arquivos representado pelo carrossel.

Os objetos do tipo diretório (“dir”) possuem a mesma sintaxe e semântica dos objetos do tipo *Service Gateway*. A diferença é que um objeto *Service Gateway* apenas representa o diretório raiz do *Service Domain*. Os objetos do tipo arquivo (“fil”) possuem como atributo, além da identificação do seu tipo, os dados relativos ao seu conteúdo.

A Figura 2 apresenta também um objeto do tipo evento (“ste”). Esse objeto é utilizado para definir tipos de eventos DSM-CC possíveis de serem descritos no fluxo de transporte. Para isso, o objeto relaciona identificadores de descritores de eventos DSM-CC a uma string (na figura, conteúdo do campo “eventName”). A utilidade desse objeto é comentada na Seção 4.

Finalmente, é importante notar, no exemplo da Figura 2, que a identificação da raiz do sistema de arquivos (diretório “weather”) é perdida, quando da geração do carrossel.

### 3. Trabalhos Relacionados

Os principais sistemas de TV digital [Morris and Chaigneau 2005] definem um esquema, um espaço de nomes e descritores, para que as aplicações interativas identifiquem, de forma apropriada, os recursos obtidos pelos receptores através do protocolo DSM-CC. Os próximos parágrafos primeiro identificam os esquemas utilizados por cada sistema, para depois analisá-los de forma crítica.

O sistema europeu DVB (*Digital Video Broadcasting*) define um mecanismo denominado *DVB Locator* para que as aplicações DVB-J ou DVB-HTML possam identificar diversos tipos de recursos, inclusive aqueles obtidos através do carrossel de objetos. Para esse último tipo de recurso, a especificação do *DVB Locator* é realizada através do esquema de URL<sup>1</sup> “dvb” [ETSI Standard 2003a]. A parte específica do esquema deve conter os seguintes identificadores: rede de onde se origina o fluxo de transporte; fluxo de transporte que está sendo recebido através do canal sintonizado; serviço sintonizado (descrito na Seção 2.1); e, por fim, o nome do recurso desejado, nesse caso, o nome de um arquivo presente no carrossel de objetos. Por exemplo, a string “dvb://1a.2b.3a/imagens/logo.gif” identifica um arquivo chamado *logo.gif*, presente no diretório *imagens*, que é a raiz de um carrossel de objetos. Como pode ser observado, os identificadores da rede, do fluxo de transporte e do serviço são separados por ponto e devem ser interpretados como valores hexadecimais [ETSI Standard 2003b].

Para desobrigar o autor das aplicações interativas de conhecer a forma de representação dos identificadores envolvidos no *DVB Locator*, o padrão do sistema europeu especifica um mecanismo que permite a descrição de forma simplificada das referências aos recursos utilizados nas aplicações interativas. No entanto, esse mecanismo está restrito à identificação de recursos de forma relativa a uma única raiz. Para isso, foram especificados um descritor para definir a localização de aplicações interativas procedurais, o *dvb\_j\_application\_location\_descriptor*, e um outro descritor para definir a localização de aplicações interativas declarativas, denominado *dvb\_html\_application\_location\_descriptor* [ETSI Standard 2003a].

Entre os campos presentes no descritor *dvb\_j\_application\_location\_descriptor*, três se destacam: *base\_directory*, *classpath\_extension* e *initial\_class*. O primeiro deve conter uma string especificando um nome de diretório, sempre a partir da única raiz “/”, que servirá como diretório base para caminhos relativos. Esse diretório é automaticamente inserido no caminho de busca por classes Java das aplicações interativas a serem chamadas. Caso o diretório base seja o diretório raiz, o campo deve conter apenas o valor “/”. O segundo campo mencionado (*classpath\_extension*), quando presente, deve possuir uma string especificando um nome de diretório relativo ao diretório base, para ser inserido no caminho de busca por classes Java das aplicações a serem chamadas. Finalmente, o campo *initial\_class* deve conter uma string especificando a classe Java por onde a aplicação deve ser iniciada.

Os principais campos do descritor *dvb\_html\_application\_location\_descriptor* são: *physical\_root* e *initial\_path*. O campo *physical\_root* deve conter uma string definindo o diretório raiz da aplicação. De forma análoga ao descritor para aplicações

---

<sup>1</sup> A sintaxe básica de uma URL é: <esquema>:<parte específica do esquema> [RFC 2005].

procedurais, caso o diretório base da aplicação seja o diretório raiz, o campo deve conter apenas o valor “/”. O campo *initial\_path* deve conter uma string definindo o caminho, relativo à raiz especificada em *physical\_root*, para o documento declarativo que deverá iniciar a aplicação interativa. Por exemplo, suponha uma transmissão que possui um diretório “/aplicação/”, contendo dois sub-diretórios: “imagens/” e “principal/”. O sub-diretório “principal/”, por sua vez, possui um documento “index.htm” e o sub-diretório “imagens” possui as imagens que servem de recursos para a aplicação declarativa. Nesse caso, o campo *physical\_root* possui como conteúdo “/aplicação/” e o campo *initial\_path* possui como conteúdo “principal/index.htm”.

O sistema americano ATSC (*Advanced Television System Committee*) [ATSC Standard 2005] também possui uma forma para identificação de recursos nas especificações ACAP. A URL definida é bem similar à do *DVB Locator*: “opcap://<sourceId>.<pid>/<path>”. Note que o esquema é definido pela string “opcap”, pois o padrão ACAP utiliza a forma de identificação definida nas especificações OCAP [CableLabs Standard 2005]. Na URL, “sourceId” identifica a rede de origem do fluxo de transporte, “pid” identifica o fluxo elementar do carrossel de objetos, e “path” identifica o recurso presente no carrossel de objetos. Além disso, o padrão ACAP define a mesma sintaxe e semântica do padrão europeu para os descritores responsáveis por localizar as aplicações procedurais e declarativas (*acap\_j\_application\_location\_descriptor* e *acap\_x\_application\_location\_descriptor*, respectivamente [ATSC Standard 2005]).

O sistema japonês ARIB (*Association of Radio Industries and Businesses*) [ARIB Standard 2002, ARIB Standard 2004] especifica a identificação de forma idêntica ao padrão do sistema europeu, através de um *ARIB Locator*, com a mesma sintaxe e semântica. A única diferença fica por conta do esquema “arib-dc” para recursos presentes no carrossel de dados japonês [ARIB Standard 2002]. Para desobrigar o autor das aplicações interativas de conhecer os identificadores envolvidos no *ARIB Locator*, o padrão do sistema ARIB especifica o descritor *arib\_j\_application\_location\_descriptor* [ARIB Standard 2004] que possui a mesma sintaxe e semântica do descritor *dvb\_j\_application\_location\_descriptor*. O padrão do sistema japonês, no entanto, não especifica esse tipo de descritor para aplicações interativas declarativas. Assim, para identificar os recursos obtidos através do carrossel de objetos, as aplicações interativas devem utilizar a URL conforme especificação do *ARIB Locator* [ARIB Standard 2002].

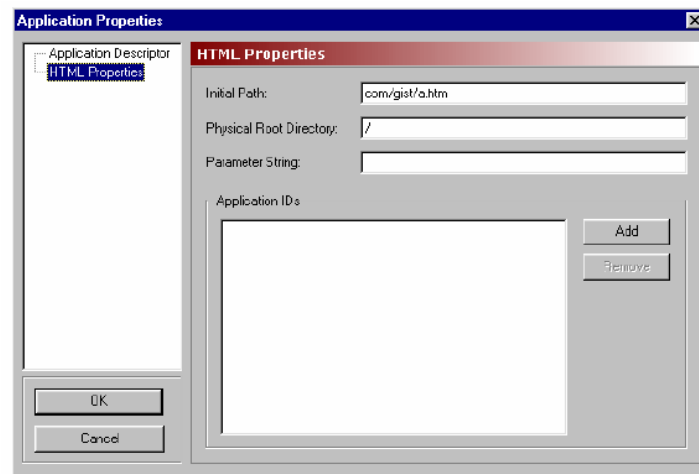
Os mecanismos especificados nos três sistemas definem como identificar recursos nos receptores (ambiente de exibição). A partir dessas especificações, são duas as abordagens existentes para traduzir a identificação dos recursos no ambiente de autoria em uma identificação dos recursos para o ambiente de exibição.

A primeira abordagem é fazer com que os autores de aplicações interativas aprendam detalhes do mecanismo de identificação de recursos do carrossel de objetos e do protocolo DSM-CC, com o objetivo de especificarem as aplicações com as URLs adequadas ao ambiente de exibição do sistema de TV digital específico. Ou, alternativamente, que todas as URLs especificadas em uma aplicação sejam automaticamente traduzidas para URLs de *Locators* identificando recursos nos carrosséis de objetos. Nesse caso, o ambiente de autoria deve manter URLs inexistentes no ambiente de autoria (as URLs só fazem sentido no ambiente de exibição), além de aumentar as chances de erros de especificação das aplicações, uma vez que a sintaxe



para localização de um recurso em um carrossel são feitas com base em códigos numéricos, conforme comentado anteriormente. Um exemplo de gerador de carrossel que oferece esse nível mais baixo de abstração para os autores de aplicações interativas é a ferramenta *dsmcc-mhp-tools*<sup>2</sup>. Note que, caso essas aplicações sejam armazenadas no cliente (receptor), para possível exibição futura, um novo sistema de arquivos deve ser gerado (nesse caso, um sistema de arquivos deve ser criado no dispositivo de armazenamento do receptor), que não tem relação alguma com o sistema de arquivos original, pois esse relacionamento foi perdido quando da tradução para as URLs dos carrosséis de objetos. Assim, uma manipulação futura desse conteúdo pelo servidor seria, se não impossível, muito difícil de ser realizada.

Uma segunda abordagem é baseada no uso dos mecanismos de descritores, definidos nos padrões dos três sistemas, conforme anteriormente descritos. Descritores gerados com auxílio de ferramentas gráficas podem ser multiplexados ao fluxo de transporte, permitindo manter no cliente as mesmas referências utilizadas no servidor, uma vez que os descritores poderiam servir de ponte para uma tradução. Como exemplo, a Figura 3 apresenta o formulário para preenchimento dos campos do descritor *dvb\_html\_application\_location\_descriptor* do gerador de carrossel de objetos da Tektronix [Tektronix 2000]. A partir do preenchimento dos campos, o gerador de carrossel tem todas as informações necessárias para a construção de um objeto, a ser enviado ao receptor, contendo a tradução das URLs relativas nas URLs dos carrosséis. No receptor, as URLs relativas seriam traduzidas para as URLs dos carrosséis, antes do início de uma aplicação ou em tempo de exibição. No entanto, tal abordagem tem um grave limitante.



**Figura 3. Gerador de Carrossel da Tektronix – Descritor DVB-HTML**

Ao definir, nos descritores discutidos, um diretório base para caminhos relativos e especificar que seu valor deve ser “/”, caso seja o diretório raiz, as especificações estão impossibilitando o uso de referências absolutas para objetos transmitidos no carrossel. Por exemplo, sem alterações nos identificadores de recursos, o mecanismo não contempla a transmissão de aplicações armazenadas no ambiente de autoria que referenciem conteúdos dispostos em dispositivos de armazenamento compartilhados da

<sup>2</sup> <http://www.linuxtv.org/dsmcc-mhp-tools.php>

rede; ou recursos dispostos em discos ou partições diferentes de uma mesma máquina, mesmo que esses recursos estejam sendo transmitidos em outro carrossel de objetos.

Como é natural no ambiente de autoria as aplicações poderem referenciar recursos de forma absoluta, para que o autor não se preocupe com URLs como as definidas pelo DVB *Locator*, seria necessário sobrecarregar o ambiente de autoria de forma a permitir que o mesmo fosse capaz de traduzir as URLs absolutas em URLs relativas, de acordo com as definições dos descritores. Para tanto, uma nova versão da aplicação interativa deve ser criada. Uma versão que possui identificadores de recursos que só fazem sentido no ambiente de exibição. Por sobrecarregar o ambiente de autoria e depender do autor para que as informações dos descritores sejam preenchidas, essa abordagem pode introduzir um retardo no processo de geração do carrossel de objetos, o que pode prejudicar o relacionamento das aplicações interativas com os programas de TV, principalmente os gerados ao vivo.

Diferente das abordagens anteriores, o middleware Ginga do Sistema Brasileiro de TV Digital propõe um novo mecanismo que não sobrecarrega a ferramenta de autoria e nem o gerador de carrossel, e, ao mesmo tempo, desobriga os autores de aplicações interativas de conhecer as idiossincrasias da identificação e resolução de recursos nos ambientes de exibição e nos sistemas de transporte. Para que essa nova abordagem seja possível, foi definido e implementado um mecanismo para enviar as informações de identificação de recursos do ambiente de autoria a serem interpretadas no ambiente de exibição. Esse mecanismo utiliza as estruturas dos comandos de edições em tempo de exibição, especificados em [Costa et al 2006]. O mecanismo proposto é detalhado na próxima seção.

#### 4. Mecanismo Proposto

O mecanismo proposto será explicado tomando como base a implementação de referência do middleware declarativo do sistema brasileiro de TV digital, bem como a implementação de referência de um servidor *datacasting* para esse mesmo sistema.

No provedor de conteúdo, quando uma aplicação interativa precisa ser transmitida via carrossel de objetos, um objeto de evento deve ser criado pelo gerador de carrossel, para informar os tipos de descritores de eventos que serão associados à aplicação. Na implementação do servidor *datacasting*, as informações de identificação de recursos das aplicações declarativas são transmitidas em descritores de eventos contendo comandos de edição [Costa et al 2006]. O middleware declarativo deve então se registrar como “ouvinte” de eventos do tipo “gingaEditingCommand”. Como consequência, um objeto de evento mapeando a string “gingaEditingCommand” em um identificador do descritor de evento DSM-CC, escolhido pelo gerador de carrossel, é transmitido no mesmo carrossel de objetos da aplicação interativa. Um ou mais descritores de eventos DSM-CC, com o mesmo identificador definido pelo objeto de evento, são então transmitidos, enviando comandos de edição sincronizados. Entre esses comandos, foram definidos um comando para carregar a aplicação, denominado “addDocument”; e um comando para adicionar estruturas lógicas (que podem possuir referências para recursos a serem apresentados pela aplicação) a uma aplicação já carregada, denominado “addNode”. Essas estruturas são especificadas através de um documento XML (*eXtensible Markup Language*) [Costa et al 2006].

Conforme descrito na Seção 2.2, os descritores de eventos DSM-CC possuem uma estrutura composta basicamente por um identificador, uma referência temporal e um campo para dados privados. O campo de dados privados é utilizado para transmitir os comandos de edição, entre eles, como já citados, os comandos “addDocument” e “addNode”. São exatamente esses os comandos usados para dar suporte ao mapeamento das estruturas de dados usadas no servidor e no cliente receptor.

Nesses comandos, um conjunto de um ou mais pares <URL, IOR> é transmitido como parâmetro. No primeiro par, a URL deve possuir o esquema “x-isdtv”, definido para o padrão brasileiro de TV digital, seguido do caminho (no ambiente de autoria) absoluto de uma aplicação interativa ou de um documento XML (contendo especificações de estruturas lógicas a serem adicionadas a uma aplicação já carregada), caso o recurso seja transmitido por carrossel. Nesse caso, uma IOR deve referenciar o objeto que transporta o recurso no carrossel de objetos. Caso contrário, a URL possui um esquema qualquer que não seja “x-isdtv” e a IOR igual a “NULL”.

Possivelmente, para contemplar todas as referências utilizadas pela aplicação (caso do comando “addDocument”) ou pelo documento XML (caso do comando “addNode”), sistemas de arquivos adicionais podem ter de ser transmitidos em outros carrosséis de objetos. Nesse caso, outros pares <URL, IOR> devem ser adicionados ao conjunto transmitido como parâmetro do comando de edição específico. A URL deve então possuir o esquema “x-isdtv”, seguido do caminho (no ambiente de autoria) absoluto da raiz do sistema de arquivos e a IOR deve referenciar o objeto (do carrossel de objetos) que representa o conteúdo desse recurso. Normalmente, essa necessidade existe quando a estrutura do sistema de arquivos que representa a aplicação não pode ser convenientemente representada por uma única árvore, conforme o exemplo apresentado na Figura 4. Nesse exemplo, explicado a seguir, a aplicação “trafficConditions.ncl” referencia o recurso “southRegion.jpg”. Entretanto, os dois conteúdos precisam ser transmitidos em carrosséis diferentes [ISO 1998]. Todo esse processo é realizado automaticamente pelo servidor *datacasting*, sem necessidade da intervenção do autor da aplicação, a quem cabe apenas o papel de criá-la e especificar o seu sistema de arquivos a ser transmitido por carrosséis de objetos.

Nos receptores, o middleware, ao receber o comando, percorre o conjunto de pares <URL, IOR> e guarda em uma estrutura de dados cada URL transmitida, relacionando-a com a URL da área de memória onde o objeto de IOR relacionada foi montado. A partir daí, para resolver as URLs existentes na aplicação interativa, o middleware consulta essa estrutura de dados, fazendo o mapeamento necessário para encontrar os recursos na área de memória do dispositivo. Note assim que as URLs presentes na aplicação do lado cliente são idênticas às aquelas presentes do lado servidor, sendo automaticamente traduzidas (mapeadas) quando qualquer referência é realizada. Note também que, diferente das soluções anteriores, não é necessário o conhecimento de estruturas complexas do carrossel de objetos, ou a necessidade da geração de uma nova versão da aplicação interativa no ambiente do servidor, com identificadores de recursos que só funcionam no ambiente de exibição. A solução é simples e eficiente, mas traz uma grande mudança de paradigma com relação aos três sistemas de TV digital citados na Seção 3: a resolução de referências não é mais realizada no lado do servidor, mas no lado do cliente.

A Figura 4 ilustra um exemplo de transmissão de aplicações interativas através do mecanismo proposto. No exemplo, o provedor de conteúdo deseja transmitir duas aplicações interativas (“weatherConditions.ncl” e “trafficConditions.ncl”) que estão armazenadas em um de seus servidores *datacasting* (Sistema de Arquivos Local, mas em discos ou partições diferentes, na Figura 4). Três carrosséis de objetos, transportando todo o conteúdo interativo, são então gerados: o carrossel de objetos apresentado na Figura 2 (Seção 2.3) e os dois carrosséis apresentados na Figura 4 (*Service Domain* = 2 e *Service Domain* = 3). Além das aplicações interativas e seus respectivos conteúdos, cada carrossel transporta ainda um objeto de eventos. O carrossel da Figura 2 relaciona o tipo de evento “gingaEditingCommand” ao identificador de evento “3” e o carrossel da Figura 4 (*Service Domain* = 2) ao identificador “4”.

O descritor de eventos “A” (Figura 4) é transmitido e identificado com o valor “4”, conforme especificado pelo objeto de evento do carrossel “2”. Nesse evento, o valor “0” é atribuído à referência temporal (campo *eventNPT*, na Figura 4, Descritor de Eventos A), fazendo com que ele seja um evento do tipo “do it now” (Seção 2.2). O campo de dados privados transporta um código definido para representar o comando “addDocument” e um conjunto formado por dois pares de URL e IOR. No primeiro par, a URL possui o esquema “x-isdtv” e o caminho local absoluto da aplicação “e:\newNclRepository\traffic”. Já a IOR localiza o objeto que representa a aplicação interativa no *Service Domain* “2”, módulo “1” e o objeto “2” do mesmo módulo e domínio. No segundo par, a URL possui o esquema “x-isdtv” e o caminho “L:\media”, enquanto que a IOR representa a imagem “southRegion.jpg” no *Service Domain* “3”, módulo “1” e o objeto “2” do mesmo módulo e domínio. Quando de posse dessas informações, o middleware guarda o caminho absoluto recebido através de cada URL transmitida, relacionando-o com a URL da área de memória onde os objetos apontados por cada IOR transmitida (no caso, a aplicação interativa) foram montados. Processo análogo é aplicado ao descritor de eventos “B” (Figura 4). Os demais campos dos descritores de eventos apresentados na Figura 4 seguem o padrão DSM-CC [ISO 1998].

Sistema de Arquivos Local	Service Domain = 2		Descritor de Eventos A
	<b>moduleId = 1</b> ... objectKey = 1 objectKind = srg 2 bindings binding #1 objectName = trafficConditions.ncl objectType = fil IOR = 2,1,2 binding #2 objectName = images objectType = dir IOR = 2,1,3 ...	<b>moduleId = 2</b> objectKey = 1 objectKind = fil data ... objectKey = 2 objectKind = ste eventList eventName = "gingaEditingCommand" eventId = 4 ...	descriptorTag = streamEventTag() descriptorLenght = descriptorLen() eventId = 4 reserved eventNPT = 0 privateDataLenght = dataLen() privateData = "0x05", "x-isdtv://E:\newNclRepository\traffic", "2,1,2", "x-isdtv://L:\media", "3,1,2"
	<b>Service Domain = 3</b>		<b>Descritor de Eventos B</b>
	<b>moduleId = 1</b> ... objectKey = 1 objectKind = srg 1 binding binding #1 objectName = southRegion.jpg objectType = fil IOR = 3,1,2 ... objectKey = 2 objectKind = fil data	descriptorTag = streamEventTag() descriptorLenght = descriptorLen() eventId = 3 reserved eventNPT = 0 privateDataLenght = dataLen() privateData = "0x05", "x-isdtv://C:\nclRepository\weather", "1,1,2"	

Figura 4. Exemplo de Uso do Mecanismo Proposto

Como apresentado, ao contrário dos mecanismos especificados nos outros sistemas de TV digital, independentemente se a aplicação interativa possui URL absoluta ou relativa, ou mesmo se o carrossel de objetos utiliza recursos de outros carrosséis, através do mecanismo proposto, a identificação de recursos é realizada nos receptores exatamente da mesma forma que no ambiente de autoria. O mapeamento para a localização física do recurso poderia ser automaticamente realizado pelo middleware. Note, como exemplo, que a aplicação interativa “trafficConditions.ncl”, além de fazer referência à imagem “southRegion.jpg”, poderia fazer uma referência à imagem “brazilianMap.jpg”, de forma relativa ou mesmo de forma absoluta. Mais ainda, essa mesma aplicação poderia referenciar um conteúdo disposto em um outro servidor, conteúdo esse carregado a partir de um outro carrossel de objetos.

É importante ressaltar que a proposta descrita neste artigo é completamente compatível com os padrões utilizados nos sistemas de TV digital citados na Seção 3. No entanto, atualizações na implementação dos respectivos middlewares seriam necessárias, uma vez que é fundamental a existência do módulo responsável por manter a estrutura de dados que relaciona as informações para identificação de recursos.

## 5. Considerações Finais

O carrossel de objetos é o protocolo comumente utilizado na transmissão de aplicações interativas nas redes de TV digital por difusão. Ao serem recebidas pelos sistemas receptores, essas aplicações são dispostas em uma área de memória do sistema cliente, ou seja, em localidades diferentes das referenciadas nas aplicações originais.

Apesar de permitirem que as aplicações consigam referenciar apropriadamente os recursos no ambiente de exibição, as especificações dos principais padrões de TV digital definem esquemas complexos para realizar a identificação desses recursos de forma absoluta, obrigando uma completa redefinição das referências usadas no ambiente de autoria para identificação dessas estruturas complexas entendidas pelo ambiente de exibição. Tal redefinição precisa ser realizada pelo autor da aplicação ou no momento da geração do carrossel de objetos. Como alternativa, mecanismos para identificação de recursos de forma relativa, também foram definidos pelos principais padrões de TV digital existentes. No entanto, tais mecanismos podem introduzir retardos no processo de geração do carrossel de objetos e sobrecarregar o ambiente de autoria, devido à possível necessidade da criação de novas versões das estruturas de arquivos das aplicações, da criação de descritores, do tratamento das tabelas PSIs e da multiplexação dessas estruturas no fluxo de transporte, conforme discutido na Seção 2.

A proposta deste artigo consiste na definição de um mecanismo para que as aplicações interativas, geradas no ambiente de autoria, consigam referenciar, tanto no próprio ambiente quanto no ambiente do cliente receptor, os recursos utilizados e transmitidos por difusão, realizando o processo de mapeamento de referências não mais no servidor, mas no cliente. Um servidor *datacasting*, localizado no provedor de conteúdos, utiliza os recursos do protocolo DSM-CC para transmitir as informações necessárias para que a identificação dos recursos seja realizada de forma única nos dois ambientes (autoria e exibição). Nos terminais de acesso, um middleware mantém uma estrutura de dados capaz de traduzir as informações de identificação transmitidas em identificações que seguem as características do ambiente do receptor. O mecanismo proposto foi validado através da implementação de referência do middleware do

Sistema Brasileiro de TV Digital, e da implementação de referência de um servidor *datacasting* para esse mesmo sistema.

Dentre as principais contribuições deste trabalho, pode-se salientar a abstração oferecida ao autor, que passa a ter apenas o papel de criar aplicações interativas e selecionar aquela a ser transmitida, não tendo mais que se preocupar com as questões relacionadas à identificação de recursos em ambientes distintos. Além disso, o mecanismo proposto permite que as aplicações interativas dispostas no ambiente de autoria sejam transmitidas sem a necessidade de alterações em seus identificadores de recursos, mesmo que esses recursos estejam localizados em sistemas de armazenamento distribuídos na rede, o que melhora o tempo para gerar o fluxo de transporte a ser transmitido.

Como trabalho futuro, pretende-se implementar o mecanismo proposto no middleware especificado pelo padrão do sistema japonês de TV digital e, nesse contexto, adaptar o servidor *datacasting* implementado para que os parâmetros sejam transmitidos diretamente no campo de dados privados dos descritores de eventos DSM-CC. Isto contribuiria em mais um passo na definição do padrão conjunto nipo-brasileiro.

## Referências

- ARIB Standard (2002). ARIB STD-B24, Version 3.2, Volume 3: Data Coding and Transmission Specification for Digital Broadcasting.
- ARIB Standard (2004). ARIB STD-B23, Application Execution Engine Platform for Digital Broadcasting.
- ATSC Standard (2005). Advanced Application Platform (ACAP), Document A/101.
- CableLabs Standard (2005). OpenCable Application Platform (OCAP), OpenCable Application Platform Specification.
- Costa et al (2006). Live Editing of Hypermedia Documents. Proceedings of ACM Symposium on Document Engineering, p. 165-172.
- ETSI Standard (2003a). Digital Video Broadcasting (DVB), Multimedia Home Platform (MHP) Specification 1.1.1, ETSI TS 102 812.
- ETSI Standard (2003b). Digital Video Broadcasting (DVB), Implementation guidelines for Data Broadcasting 1.2.1, ETSI TR 101 202.
- ISO (1998). ISO/IEC 13818-6. Information technology – Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC.
- ISO (2000). ISO/IEC 13818-1. Information technology – Generic coding of moving pictures and associated audio information - Part 1: Systems.
- Morris, S. and Chaigneau, A. S. (2005). Interactive TV Standards. Focal Press, Elsevier.
- OMG Specification (2004). Common Object Request Broker Architecture (CORBA/IIOP).
- RFC (2005). Request for Comments: 3986. Uniform Resource Identifier (URI): Generic Syntax.
- Tektronix (2000). AD951A/AD953A MPEG System User Manual – Carousel Generator.