

10 Conclusões

Entre as principais questões envolvidas no planejamento, desenvolvimento e implementação de middlewares declarativos para TVD interativa está a economia de recursos computacionais e o suporte à evolução do middleware em tempo de execução. As soluções baseadas em componentes de software vêm se mostrando atraentes para prover um alto grau de adaptabilidade e um maior controle do uso de recursos computacionais. O desenvolvimento da implementação de referência do middleware Ginga (seus subsistemas Ginga-NCL e Ginga-CC) baseado em componentes teve esse foco.

A confiabilidade exigida em um middleware para TVD também é um dos principais requisitos de seu projeto. A exibição do conteúdo, de aplicações interativas não pode ser comprometida devido à ocorrência de falhas na execução do middleware. No entanto, em todas as estruturas de dados necessárias para o plano de apresentação de aplicações interativas, definidas por Costa et al. (Costa, 2006a) em especial para o middleware Ginga-NCL, não houve a preocupação com o suporte a recuperação de falhas ocorridas no middleware ou mesmo nas aplicações para esse sistema. O desenvolvimento da implementação de referência do middleware Ginga leva em conta requisitos de recuperação de falhas.

A manutenção do sincronismo em uma aplicação de TV digital deve ser o requisito principal de um middleware. A provisão de um alto grau de adaptabilidade, um maior controle do uso de recursos computacionais e a provisão de mecanismos de recuperação de falhas não podem comprometer o requisito principal. Na arquitetura Ginga proposta nesta tese, e na implementação de referência baseada nesta arquitetura, todo o controle do momento de carga de componentes e do momento de execução de mecanismos de recuperação é definido, de modo a não causar retardos que possam influenciar no sincronismo dos vários conteúdos de mídia das aplicações.

No desenvolvimento e apresentação das aplicações, um ponto importante no cenário da TVD é o suporte a edições das aplicações enquanto são exibidas.

Considerando que os ambientes de autoria, transmissão e exibição consistem em um sistema distribuído, a principal dificuldade é possibilitar a coerência total entre as especificações na autoria e na exibição.

O carrossel de objetos é o protocolo comumente utilizado na transmissão de aplicações interativas nas redes de TVD 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 TVD definem esquemas complexos para realizar a identificação desses recursos de forma absoluta, obrigando a redefinição das referências usadas no ambiente de autoria para identificação das estruturas entendidas pelo ambiente de exibição. Tal redefinição precisa ser realizada pelo autor da aplicação ou no momento da geração do carrossel de objetos. Como alternativas, alguns 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.

A proposta discutida nesta tese permite que aplicações geradas no ambiente de autoria consigam referenciar, tanto no próprio ambiente quanto no ambiente do cliente receptor, os recursos utilizados e transmitidos, realizando o processo de mapeamento de referências não mais no servidor, mas no cliente, de forma completamente escondida dos autores das aplicações. 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 receptores, 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.

A preservação do sincronismo intermídia em aplicações que referenciam conteúdos transmitidos de forma intercalada com outros conteúdos, em um mesmo fluxo contínuo, não é um problema trivial. Para resolução desse problema, típico de aplicações para TVD, soluções são discutidas nesta tese.

Em um cenário típico de TVD, uma aplicação pode referenciar o fluxo de vídeo principal que, usualmente é intercalado por propagandas e seguido por outros programas (conteúdos). Mais ainda, ao usuário telespectador é permitido receber e executar a aplicação qualquer que seja o tempo de sintonização de um canal. Agravando o fato, a sintonização de um canal pode ser um retorno após uma breve visita a outro canal, quando, então, não é desejável que a aplicação, anteriormente iniciada, seja finalizada e recomeçada sem levar em conta a sua história passada. Todas essas características tornam mais complexa a preservação do sincronismo e o controle do ciclo de vida das aplicações.

Em alguns dos principais sistemas de TVD, como apresentado, o controle das bases temporais ou é insuficiente, ou demanda que os autores de aplicações (ou sistemas de autoria) conheçam os detalhes da implementação do sincronismo, sejam os controladores da execução da aplicação e vislumbrem toda a infraestrutura de serviços de transporte assíncronos do sistema de TVD.

Esta tese discutiu e propôs um controle para o ciclo de vida das aplicações, em que o conteúdo de seus objetos de mídia são tratados de modo totalmente opaco ao autor das aplicações. Do autor da aplicação não é exigido nenhum conhecimento como as bases temporais são sinalizadas, como elas são associadas aos objetos transmitidos em fluxos elementares e como são identificados os pontos em que o conteúdo desses objetos se entrelaçam em um fluxo elementar.

Ginga é o middleware padrão para sistemas de TVD terrestre e também para serviços IPTV. Suas estruturas de dados devem poder ser transportadas nos mais diversos sistemas de transportes assíncronos, de forma eficiente e leve. Para tanto várias estruturas de dados e metadados foram definidas e como essas estruturas pode ser mapeadas de forma eficiente em diversos sistemas de transporte.

Terminando este Capítulo, repetindo a introdução desta tese, as principais contribuições encontradas neste trabalho podem ser assim resumidas:

- Projeto de uma arquitetura para o middleware Ginga-NCL, com a possibilidade de implementação baseada em componentes de software, com um alto grau de

adaptabilidade em tempo de execução e um maior controle sobre os recursos computacionais utilizados;

- Projeto de um plano de recuperação de componentes capaz de prover resiliência tanto no Ginga-NCL quanto nas aplicações que utilizam esse sistema;
- Novo método de identificação de bases temporais, preservando o sincronismo das aplicações sem que seus autores tenham necessidade de conhecer os detalhes de sua implementação;
- Definição de Comando de Edição que entre outras facilidades permite:
 - O controle do ciclo de vida de aplicações, com novas ações de controle declarativas, inexistentes nas propostas correntes, com maior eficiência e sem impor ao autor das aplicações o overhead mental de criação de mecanismos complexos de controle;
 - O disparo de uma aplicação, independente do momento em que um telespectador inicia a recepção da transmissão, preservando todos os sincronismos espaço-temporais que elas determinam;
 - A manutenção do status de uma aplicação, permitindo que mudanças de sintonização realizadas não prejudiquem seu fluxo quando resintonizadas;
 - O suporte aos autores de aplicações na identificação de recursos, permitindo que não tenham que se preocupar com as questões relacionadas à localização dos recursos em ambientes distintos de autoria, transmissão e exibição, além de permitir que as aplicações interativas dispostas no ambiente de autoria sejam transmitidas sem a necessidade de alterações em seus identificadores de recursos, diminuindo o retardo na geração do fluxo de transporte a ser transmitido;
 - A realização de modificações ao vivo (em tempo de exibição) nas aplicações de modo coerente, preservando todos os relacionamentos especificados pelo autor, incluindo aqueles que definem a estruturação lógica da aplicação;
- Definição de uma ampla infra-estrutura de suporte a serviços de transporte assíncronos de forma extensível e independente do sistema de distribuição de TVD, por meio da definição de estruturas de dados e metadados que podem ser

adequados à codificação de transporte específica de cada sistema de distribuição;

- Como prova de conceito: implementação de referência para o middleware Ginga (seus subsistemas Ginga-NCL e Ginga-CC) seguindo todos os pontos anteriormente mencionados.

A implementação de referência do middleware Ginga-NCL encontra-se disponível para download sob licença livre na Comunidade Ginga (www.softwarepublico.gov.br).

Na implementação atual, parte da estrutura de controle é baseada no HTG e parte em estruturas internas dos objetos NCM gerados quando da conversão (módulo Converter da arquitetura Ginga-NCL) da aplicação NCL. A utilização dessas duas estruturas é histórica no desenvolvimento dos players NCL e não é a ideal. O modelo NCM possui várias estruturas complexas que estão em um nível de abstração longe do entendimento da máquina de execução. Um novo perfil NCL, tão expressivo quanto o perfil *Full Language* da linguagem, mas sem nenhum dos “açucares sintáticos” de NCL encontra-se atualmente em desenvolvimento. Tal perfil servirá de base para uma nova implementação, ainda mais eficiente e resiliente, onde todas as vantagens da arquitetura proposta poderão ser exploradas mais profundamente.

Outro trabalho que se pretende realizar é a investigação em mais detalhes do possível impacto de desempenho e do ganho de novos recursos ao se implementar a arquitetura apresentada neste trabalho com base em uma infra-estrutura de componentes como OpenCom, Fractal e FlexCM. Pretende-se também analisar os impactos das duas configurações (MONO e COMP) da implementação de referência em situações que existir uma partição de swap habilitada.

Apesar de proposto para o Ginga-NCL, o plano de recuperação pode ser aplicado a outros middlewares de TVD. Pretende-se aplicar no futuro a arquitetura do plano de recuperação ao exibidor XHTML integrado ao Ginga-NCL. Além disso, é prevista a integração dos recursos oferecidos pelas cadeias temporais e as estruturas do HTG ao plano de recuperação para refinamento das estratégias de recuperação proativa e reativa. Definições de políticas de recuperação também são vislumbradas, bem como de outras falhas e ações como, por exemplo, a utilização

de algoritmos de ajuste elástico (Maranhao, 2006) quando da perda de sincronismo.

Ainda como trabalho futuro pretende-se desenvolver um framework onde o mapeamento para os comandos de edição ocorra de acordo com a estrutura da linguagem declarativa proposta por um determinado modelo hipermídia, podendo aplicar os conceitos dos comandos de edição NCL em outros ambientes declarativos de TVD.

Outro trabalho futuro proposto consiste na implementação do mecanismo de identificação de recursos proposto no middleware especificado pelo padrão do sistema japonês de TVD 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 nipo-brasileiro ISDB-T.

O desenvolvimento de novos componentes para a especialização da implementação Ginga-NCL para ambientes IPTV está em andamento, buscando a conformidade com a recomendação ITU-T H.761 (ITU-T, 2009). Os sistemas de transporte oferecidos serão o carrossel de objetos DSM-CC e o protocolo FLUTE. De fato, FLUTE é um protocolo de transporte para aplicações e dados mais direcionado ao ambiente IPTV, uma vez que oferece pronto suporte a comunicação *multicast*, independe da existência de multiplexação dos dados com o fluxo audiovisual principal, além de possuir mecanismos para tolerância a congestionamento de rede e correção de erros.

A partir da especialização da codificação de dados para o transporte em IPTV, pode-se vislumbrar novas frentes de trabalho, como a adequação desse transporte em serviços *peer-to-peer* TV (P2PTV) (Simpson, 2008). Nesse contexto, não somente o fluxo audiovisual principal de um canal P2PTV pode estar segmentado (e de alguma forma redundante) em diversos pontos da rede, mas também as aplicações de TVD. O tratamento de tal distribuição granular do conteúdo interativo compreende um trabalho futuro promissor.