

# 1 Introdução

Um sistema de TV Digital (TVD) pode ser definido, resumidamente, como um conjunto de especificações que determinam as tecnologias envolvidas na transmissão de conteúdo pelas emissoras (ou provedores de conteúdo) e no tratamento de conteúdo nos ambientes de recepção dos telespectadores. Nesse cenário, o suporte à interatividade é realizado por uma camada intermediária de software, ou middleware, posicionada, no ambiente de recepção, entre o código das aplicações interativas e a infra-estrutura de execução (plataforma de hardware e sistema operacional).

O projeto e implementação de um middleware para receptores de sistemas de TVD trazem alguns desafios. Entre eles está a eficiência na gerência de recursos utilizados. Geralmente, um dispositivo receptor possui hardware especializado para a exibição de conteúdo televisivo e apresenta forte limitação na quantidade de recursos computacionais: processador de baixo custo para execução das tarefas das aplicações interativas e do sistema operacional (SO), bem como pouca memória para uso dessas aplicações e do SO.

Outro aspecto importante no desenvolvimento de um middleware para sistemas de TVD é o suporte à sua atualização em tempo de execução, ou sua evolução dinâmica. O suporte à evolução dinâmica permite a integração de novas funcionalidades, substituição de funcionalidades existentes e redefinições arquiteturais, originadas das potenciais mudanças de requisitos não previstas no projeto inicial (Barbosa, 2006).

A confiabilidade exigida nessa camada de software consiste em outro requisito essencial. A exibição do conteúdo, incluindo a apresentação de aplicações interativas, não deve ser comprometida quando da ocorrência de falhas no middleware. Segundo Koren e Krishna (Koren, 2007), as diversas ferramentas e técnicas criadas para reduzir o número de falhas nos sistemas de software não são suficientes para garantir a confiabilidade: é preciso construir sistemas que reconheçam a existência de falhas e incorporem técnicas para tolerar essas falhas,

ou mesmo recuperar-se delas como em um sistema resiliente, enquanto ainda estiverem oferecendo um nível aceitável de serviço.

Durante a exibição de uma aplicação interativa, é desejável que o ambiente de recepção seja, ainda, dotado de recursos para permitir que essa aplicação seja modificada, de forma que as modificações sejam automaticamente refletidas na apresentação corrente. Em programas de TV ao vivo e programas modificados por retransmissoras, essa facilidade de edição das aplicações durante sua exibição é bastante atraente.

Em uma aplicação interativa de TVD, além dos fluxos de áudio principal (ou primário) e vídeo principal (ou primário), outros objetos de mídia podem ter suas exibições sincronizadas no tempo e no espaço.

Os fluxos de áudio e de vídeo principal podem ser transmitidos por *multicast*, como em sistemas IPTV, ou mesmo por difusão, como acontece nos sistemas de TVD por difusão terrestre. Os demais objetos de mídia da aplicação, bem como o documento de especificação responsável por relacionar no tempo e no espaço os vários objetos de mídia, podem ser obtidos do mesmo serviço (ou da mesma rede) em que são transportados os fluxos de áudio e de vídeo principal, ou de outras redes, como por exemplo, o canal de retorno (ou de interatividade) em um sistema de TVD terrestre.

Independente da forma de transmissão, a localização dos vários objetos de mídia de uma aplicação é definida pelo documento de especificação da aplicação, criado no que chamamos de fase de autoria. Como a localização conhecida pelo sistema de autoria é diferente da localização dos mesmos objetos quando transportados (fase de transferência) e entregues a um receptor para exibição (fase de apresentação), metadados são necessários para que a sintaxe de autoria de uma aplicação possa ser compreendida em seu destino final.

O controle do ciclo de vida das aplicações, principalmente as geradas ao vivo e aquelas relacionadas temporalmente com o fluxo audiovisual principal, também requer o uso de metadados para o transporte de comandos de controle, incluindo comandos que permitem a edição da aplicação.

Um ponto importante no ciclo de vida de uma aplicação TVD é o estado inicial de sua apresentação. Como na TV analógica, no cenário da TVD por difusão, telespectadores podem começar a assistir um programa já iniciado. Mais ainda, telespectadores podem trocar de canal e, conseqüentemente, sair e entrar

em aplicações interativas já em andamento. O tratamento de objetos de mídias contínuas, cujo fluxo pode ter iniciado antes do início da exibição da aplicação interativa que os contém, se faz necessário e é outro requisito de grande importância.

### 1.1. Objetivos e Contribuições

O presente trabalho tem como objetivo o desenvolvimento do middleware declarativo para sistemas de TVD denominado Ginga-NCL, de forma a conciliar eficiência, resiliência e flexibilidade no suporte à apresentação de aplicações interativas especificadas na linguagem NCL (*Nested Context Language*) (Soares, 2006). Para satisfazer esse objetivo, várias questões, algumas ainda em aberto e outras com soluções pouco eficientes, precisam ser resolvidas. Com esse foco, os seguintes objetivos específicos/requisitos foram levantados:

1) **Definição da arquitetura de referência para o middleware declarativo Ginga-NCL**, levando em consideração:

- a. **Gerência de Recursos e Evolução Dinâmica:** especificação e desenvolvimento de uma arquitetura em componentes, de forma que conjuntos de funcionalidades utilizem recursos do dispositivo receptor apenas enquanto forem necessários. Além disso, a arquitetura deve suportar a evolução dinâmica do middleware, considerando que as atualizações devem ser realizadas sem a necessidade de interromper o funcionamento do middleware e sem interromper a apresentação de aplicações NCL;
- b. **Plano de Recuperação:** entre todas as estruturas de dados necessárias para o plano de apresentação de aplicações interativas, definidas por Costa et al. (Costa, 2006a), não houve a preocupação com o suporte a recuperação de falhas. Para esse suporte, é necessário também a especificação e desenvolvimento de um plano de escalonamento de recuperação.

- 2) **Edição e Controle do Ciclo de Vida das aplicações NCL, incluindo em tempo de exibição**, uma abordagem que deve atender os seguintes pontos:
- a. **Comandos de edição:** especificação e desenvolvimento do suporte aos comandos de edição, que possuem como proposta inicial as discussões realizadas por Costa et al. (Costa, 2006b);
  - b. **Identificação de recursos:** especificação e desenvolvimento da gerência de localização de recursos de forma a permitir o uso da mesma sintaxe de autoria em fases distintas: fase de autoria e fase de apresentação;
  - c. **Ciclo de Vida de aplicações NCL:** integração entre os serviços TVD e a apresentação de aplicações NCL, de forma a considerar o uso de bases temporais contextualizadas (ISO, 1998) e comandos de controle por metadados TVD como, por exemplo, eventos de sincronismo DSM-CC (ISO, 1998) e tabelas de informação de serviço (*Service Information – SI*) (Morris, 2005);
  - d. **Sistema de Transporte:** definição de uma infra-estrutura de suporte a serviços de transporte assíncronos de aplicações interativas e comandos de controle, de forma extensível e independente do sistema de distribuição de TVD.

Tendo por base o objetivo geral definido e atendendo aos requisitos levantados, as seguintes contribuições são apresentadas nesta tese:

#### **A partir do objetivo específico 1:**

- Implementação do middleware Ginga-NCL, realizada como prova de conceito da especificação e desenvolvimento da arquitetura do middleware declarativo para suporte à apresentação de aplicações NCL. O desenvolvimento tornou-se a implementação de referência da recomendação ITU-T para serviços IPTV (ITU-T, 2009; Soares, 2010);
- Validação de modelos propostos pela especificação NCL através da implementação de referência, contribuindo para o refinamento dos mesmos, entre os quais se podem citar o suporte a múltiplos

dispositivos (Soares, 2009b) e a objetos declarativos (Moreno, 2009b);

- Desenvolvimento de uma arquitetura orientada a componentes de software para middlewares com foco em TVD. Até onde sabemos, este é o primeiro trabalho que reporta a aplicação de uma arquitetura baseada em componentes de software na construção de um middleware declarativo para TVD (Moreno, 2010c);
- Desenvolvimento de um plano de recuperação capaz de introduzir resiliência não apenas no middleware Ginga-NCL, mas também na apresentação de aplicações interativas que esse middleware realiza (Moreno, 2010b).

### **A partir do objetivo específico 2:**

Como já mencionado, as soluções oferecidas (Morris, 2005) pelos middlewares padrões para TVD apresentam algumas limitações importantes. Soluções alternativas foram propostas para o middleware Ginga-NCL, contornando as limitações presentes em outros sistemas.

- Os comandos de edição de Costa et al. (Costa, 2006b), definidos para controle do escalonamento de apresentação, foram refinados, evoluindo da proposta inicial de controle do escalonamento de apresentações para o suporte a edição de aplicações em tempo de execução, que foram incorporadas nas especificações das normas Ginga-NCL (ABNT, 2009; ITU-T, 2009);
- Uso da mesma sintaxe de autoria na fase de apresentação (Moreno, 2007), apresentando uma nova proposta para o mapeamento da localização dos vários objetos de mídia de uma aplicação;
- Integração entre o acesso às bases temporais e o suporte a entrada em qualquer ponto de uma aplicação (Moreno, 2008), contornando as limitações no controle de ciclo de vida das aplicações;
- Especificação e desenvolvimento de alternativas de transporte de dados, mais apropriadas para sistemas particulares como, por exemplo, TVD por difusão, IPTV, etc. (Moreno, 2010a).

## 1.2. Motivação

Historicamente, a TV aberta brasileira tem grande importância não só como fonte de entretenimento, mas como um veículo da comunicação social. Segundo o IBGE<sup>1</sup>, no ano de 2005, 92% dos domicílios, distribuídos pelo extenso território nacional, possuíam TV. Nesse mesmo ano, em vez de adotar um sistema de TVD existente, o governo do Brasil enxergou a necessidade da concepção de um sistema próprio, capaz de atender às demandas brasileiras, favorecendo os objetivos políticos, sociais e econômicos do país. Iniciavam-se os esforços para a definição do Sistema Brasileiro de TV Digital por difusão Terrestre (SBTVD-T), uma oportunidade de explorar a relevante penetração da TV nos lares brasileiros como uma nova ferramenta para inclusão digital.

O decreto (PRCC, 2010) que dispõe sobre a implantação do SBTVD-T define que esse sistema deve possibilitar a transmissão digital em alta definição (*High Definition TV* – HDTV) e em definição padrão (*Standard Definition TV* – SDTV); a transmissão digital simultânea para recepção fixa, móvel e portátil; e a interatividade. Entre os principais efeitos da implantação são observadas a atualização e revitalização da indústria eletrônica e do setor de radiodifusão, bem como a promoção da inclusão digital.

A linguagem declarativa NCL (*Nested Context Language*) (Soares, 2006) tornou-se a solução para o suporte declarativo do SBTVD-T, não apenas pela facilidade que ela oferece para especificar os relacionamentos espaciais e temporais entre os objetos de mídia, mas também pelo suporte declarativo a adaptações de conteúdo e de formas de apresentação de conteúdo, o suporte declarativo a múltiplos dispositivos de exibição e a edição/produção das aplicações em tempo de exibição.

No início dos esforços deste trabalho, duas máquinas de apresentação (comumente denominadas formatador nos sistemas hipermídia) para documentos NCL se destacavam, o Formatador HyperProp (Rodrigues, 2003) e a máquina Maestro (Moreno, 2006), ambos desenvolvidos pelo laboratório TeleMídia.

O Formatador HyperProp, inicialmente desenvolvido como uma ferramenta hipermídia para Web, recebeu diversas atualizações na tentativa de suportar o

---

<sup>1</sup> Informação obtida através das séries estatísticas disponíveis em [www.ibge.gov.br](http://www.ibge.gov.br).

perfil de TVD da linguagem NCL (Soares, 2006). As versões criadas do Formatador HyperProp foram implementadas na linguagem Java e, naturalmente, utilizavam JMF (*Java Media Framework*) para exibição de conteúdo de mídia contínua. No entanto, a dificuldade de atender aos requisitos de um sistema de TVD, que apresenta um cenário de plataformas com poucos recursos e outras idiossincrasias (Moreno, 2006), fez com que essa ferramenta fosse descontinuada em 2007.

A máquina Maestro (Moreno, 2006) foi desenvolvida tendo por base o Formatador HyperProp. As versões do Maestro foram implementadas na linguagem C++ e desenvolvidas com foco na TVD. No entanto, apesar de modelado como middleware em (Moreno, 2006), a máquina Maestro não chegou a incorporar as funcionalidades de middleware em um ambiente de difusão terrestre e serviu apenas como prova de conceito. Descontinuada no segundo semestre de 2006, parte da implementação da máquina Maestro foi aproveitada para iniciarem os esforços no desenvolvimento do middleware declarativo Ginga-NCL.

O middleware Ginga-NCL e sua linguagem declarativa NCL foram adotados pelo SBTVD-T em 2007. No início de 2009, Ginga-NCL e NCL tornaram-se parte dos padrões ISDB-T<sub>B</sub> (o antigo padrão japonês, mas agora incorporando as inovações brasileiras, renomeado como *International Standard for Digital Broadcasting*) e parte da recomendação ITU-R BT 1699 (ITU-R, 2009). Ainda no início de 2009, Ginga-NCL e NCL tornaram-se a primeira tecnologia padronizada para aplicações multimídia para serviços IPTV, por meio da Recomendação ITU-T H.761 (ITU-T, 2009). Os trabalhos desenvolvidos nesta tese contribuíram na definição dessas várias inovações brasileiras.

### **1.3. Organização da Tese**

O restante da tese encontra-se organizado da seguinte forma. O Capítulo 2 apresenta algumas discussões sobre os principais trabalhos relacionados. No Capítulo 3 é discutida a aplicação dos conceitos do desenvolvimento orientado a componentes de software na arquitetura do middleware declarativo Ginga-NCL, de forma a prover eficiência na gerência de recursos, bem como mecanismos para

a evolução dinâmica do middleware. O Capítulo 4, por sua vez, discorre sobre a proposta de um plano de recuperação para o middleware declarativo Ginga-NCL, capaz de introduzir resiliência à apresentação de aplicações NCL. A facilidade de edição das aplicações durante sua exibição é discutida no Capítulo 5. Já o Capítulo 6 apresenta um mecanismo de identificação de recursos para aplicações interativas em redes de TVD. No Capítulo 7 é discutido como os comandos de edição podem ser utilizados para o controle do ciclo de vida das aplicações NCL. O Capítulo 8 apresenta as várias alternativas de transportes definidas para o middleware Ginga-NCL. Os trabalhos relacionados são revisitados no Capítulo 9, que realiza uma análise comparativa entre esses trabalhos e as soluções propostas neste documento. O Capítulo 10 encerra a tese, descrevendo as conclusões obtidas a partir de todo trabalho realizado e os possíveis trabalhos futuros.