

# 1 Introdução

Uma aplicação de TV Digital (TVD) comumente é composta por um vídeo principal associado a outros objetos (aplicações, imagens, vídeos, textos etc.), que são transmitidos em conjunto possibilitando um programa de TV não-linear. Os conteúdos possuem entre si relacionamentos de sincronismo no tempo e espaço, o que é descrito no contexto da especificação de uma aplicação de TVD.

Usualmente os sistemas de TVD dão suporte à especificação de aplicações sob os paradigmas imperativo ou declarativo. Ginga-J (Souza, 2007) e OCAP-RI (*OpenCable Application Platform – Reference Implementation*) (OCAP-RI, 2011) são exemplos de sistemas que dão suporte a especificações sob o paradigma imperativo. LAsER (*Lightweight Application Scene Representation*) (ISO, 2006), BML (*Broadcast Markup Language*) (B24, 2004) e Ginga-NCL (Soares, 2007; Souza, 2007) são exemplos de sistemas que dão suporte a especificações sob o paradigma declarativo.

As especificações de aplicações declarativas de TVD são determinadas por meio de *documentos*. Esses documentos têm como função principal comunicar quais objetos de mídia compõem uma aplicação e como esses objetos se relacionam no tempo e espaço. Os documentos de especificação são expressos em alguma linguagem declarativa por parte de um produtor de conteúdo, transmitidos juntamente com os objetos de mídia que a compõem, para por fim serem interpretados por agentes de usuário em um receptor (cliente) de TVD. Esse agente de usuário pode ser, por exemplo, um formatador que controla a exibição dos objetos de mídia de uma aplicação a partir de relacionamentos especificados no documento. A fase de preparação desses documentos é chamada comumente de *tempo de autoria* e é feita em ambientes ou ferramentas próprias para esse fim. A exibição do documento ocorre em uma fase distinta, executada pelo agente de usuário, a qual é chamada de *tempo de apresentação*.

Alguns sistemas de TVD dão suporte à geração dinâmica do conteúdo de aplicações e à edição dinâmica de documentos. Isso significa que o documento

pode ser especificado de tal forma a prever modificações em si mesmo em tempo de apresentação. O dinamismo em tais documentos torna menos clara a separação entre tempo de autoria e de apresentação. Isso porque os conteúdos dos documentos – objetos de mídia e relacionamentos entre objetos de mídia – podem ser dinamicamente inseridos, removidos ou modificados em tempo de apresentação. Esse recurso de geração dinâmica de documentos permite o desenvolvimento de uma nova gama de aplicações em que parte do conteúdo é concebida apenas em tempo de apresentação quando, por exemplo, é o momento de o usuário interagir com a aplicação ou apenas quando se conhece o perfil do usuário da aplicação, ou ainda quando a aplicação obtém dados atualizados.

Neste trabalho é demonstrado como é possível estabelecer um padrão de arquitetura a ser seguido por um tipo especial de aplicações TVD que têm conteúdo gerado dinamicamente. O foco desta dissertação está na definição dessa arquitetura, e em como ela pode contribuir para facilitar o desenvolvimento de aplicações que se utilizam da geração dinâmica de conteúdo.

Para ilustrar o padrão de arquitetura proposto, é oferecida uma implementação baseada em NCL (*Nested Context Language*) (ITU-T, 2009), linguagem de especificação de documentos para TV digital. NCL é a linguagem declarativa padrão do Ginga (Soares, 2007; Souza, 2007), *middleware* do Sistema Nipo-Brasileiro de TV Digital (ISDB-TB) e recomendação ITU-T (ITU-T, 2009) para IPTV. Seu subsistema Ginga-NCL fornece suporte à geração dinâmica de conteúdo (Costa, 2006).

A Seção 1.1 introduz o que são aplicações de TVD dinâmicas e cita dois tipos dessas aplicações, dando ênfase às denominadas *aplicações que se recriam dinamicamente*. Em seguida, a Seção 1.2 apresenta a motivação para se definir uma arquitetura para esse tipo de aplicação. Essa arquitetura, descrita com maiores detalhes no Capítulo 3, se baseia no método de autoria orientado a *templates* (Soares Neto, 2010), e um dos objetivos da Seção 1.2 é mostrar como os conceitos utilizados nesse método podem ser úteis no processo de desenvolvimento de uma aplicação que se recria dinamicamente. Para encerrar este capítulo, a Seção 1.3 apresenta os objetivos e a Seção 1.4 apresenta a estrutura da dissertação.

## 1.1. Aplicações de TVD dinâmicas

Aplicações de TVD dinâmicas são caracterizadas por possuírem conteúdos que podem ser gerados ou modificados em tempo de apresentação. Elas são importantes porque:

- Seus objetos de mídia e relacionamentos hipermídia podem ser gerados em aplicações de TVD para programas transmitidos ao vivo. Por exemplo, guias eletrônicos de programação podem ser gerados em tempo real;
- Oferecem a possibilidade de postergar a especificação de parte do conteúdo que envolva processamento complexo e que não precise ser realizado antes da ocorrência de determinados eventos. Por exemplo, a geração de um menu de vídeos relacionados semanticamente a outro vídeo só precisa ser feita após uma ação de seleção pelo usuário;
- Permitem personalizar os conteúdos conforme as características dos usuários da aplicação, dados das interações do usuário etc.

O sistema ISDB-TB e seu *middleware* Ginga, mais especificamente seu ambiente declarativo para aplicações NCL, foram estabelecidos na seção anterior como a plataforma de TVD em que esta dissertação se baseia. Ginga dá suporte à especificação de aplicações NCL dinâmicas por meio dos seguintes métodos de geração de conteúdo dinâmico:

### 1. *Objetos NCLua*

Ginga-NCL dá suporte a objetos cujo conteúdo consiste em código imperativo escrito na linguagem Lua, chamados de objetos NCLua. A API NCLua *Canvas* oferece funcionalidades para que conteúdos possam ser apresentados em uma aplicação. Utilizando essa API, esses conteúdos podem ser incluídos dinamicamente na exibição de uma aplicação durante a execução de um objeto NCLua;

## 2. *Comandos de edição externos aos clientes*

As aplicações de TVD são transmitidas para o cliente por meio de um sinal digital com vários tipos de fluxos multiplexados, tais como os fluxos de vídeo principal, áudio principal, dados (outros objetos de mídia, sistemas de arquivos etc.) etc. O agente de usuário no cliente demultiplexa o sinal para exibir a aplicação. No ISDB-TB, a transmissão dos dados da aplicação segue um protocolo de carrossel de objetos DSM-CC (ISO/IEC, 1998), que consiste em enviar ciclicamente esses dados para o cliente a fim de que sejam remontados conforme são recebidos. Além desses fluxos, é possível enviar também fluxos de eventos (*stream events*). Objetos de eventos (*event objects*) são enviados por provedores de conteúdo em fluxos de eventos (*stream events*) do carrossel de objetos, que podem conter comandos destinados a realizar alterações na aplicação e que são tratados pelo cliente. No Ginga, esses comandos permitem, por exemplo, a adição de novos objetos (comando *addNode*), relacionamentos (comando *addLink*) etc. (Costa, 2006);

## 3. *Comandos de edição provenientes de objetos NCLua*

O formatador NCL se comunica por tratamento de eventos com objetos NCLua da aplicação. Uma das classes de eventos que existem nessa comunicação é o *edit*. Eventos do tipo *edit* gerados por objetos NCLua agem semelhantemente aos comandos de edição externos, informando ao formatador alterações a serem realizadas na aplicação. A possibilidade de gerar eventos da classe *edit* permite a adição de novos conteúdos (objetos de mídia e relacionamentos entre os mesmos) na aplicação.

## 4. *Elementos switch e descriptorSwitch*

O conteúdo de uma aplicação NCL pode ser adaptado em tempo de apresentação por meio de elementos *switch* e a forma de apresentação desse conteúdo pode ser adaptado por meio de elementos *descriptorSwitch*.

Um elemento *switch* é uma composição que contém objetos alternativos, ou seja, dentre os quais apenas um é selecionado. Esses objetos podem ser de mídia, de contexto e outros *switches*. O nó selecionado em um *switch* é determinado a partir da avaliação de regras de mapeamento definidas dentro do próprio *switch* por elementos *bindRule*.

Da mesma forma que se utiliza o elemento *switch* para selecionar um conteúdo conforme uma regra, o elemento *descriptorSwitch* serve para selecionar um descritor conforme uma regra. Assim como no *switch*, são utilizados elementos *bindRule* para ligar uma regra ao descritor correspondente.

##### 5. *Objetos HTML referenciando páginas dinâmicas*

HTML é uma linguagem com suporte à exibição de conteúdo dinâmico. Várias linguagens de *script* podem ser embutidas em HTML com esse propósito, como ECMAScript (ECMA, 1999), JavaScript (Flanagan, 2011). O suporte a conteúdos dinâmicos é obtido diretamente, uma vez que NCL permite a exibição de objetos HTML.

Aplicações de TVD dinâmicas possuem objetos de mídia especiais que geram outros objetos de mídia e relacionamentos dinâmicos na aplicação. Elas iniciam um processo de geração de parte do conteúdo da aplicação a partir da ocorrência de algum tipo de evento. Em NCL, por exemplo, esses objetos de mídia especiais são os objetos NCLua, que quando executados podem modificar o conteúdo de documentos por meio de eventos de edição enviados ao formatador NCL.

Como um caso particular, há aquelas aplicações que reestruturam seu próprio conteúdo completamente, denominadas nesta dissertação como *aplicações que se recriam dinamicamente*. Essas aplicações iniciam um processo de recriação completa de seu conteúdo a partir da ocorrência de algum tipo de evento. Nesse caso, ao fim do processo de geração de conteúdos, as aplicações recriadas devem permanecer conforme um padrão de composição, ou *template* de composição. Para a especificação desse tipo de aplicação dinâmica, são utilizados o primeiro e terceiro métodos mencionados anteriormente (objetos NCLua e comandos de edição provenientes de objetos NCLua).

Os *templates* de composição são usados na arquitetura proposta nesta dissertação para definir restrições sobre as aplicações dinâmicas que se recriam dinamicamente. A próxima seção inicia uma discussão acerca das aplicações que se recriam dinamicamente, e a conveniência da arquitetura dessas aplicações utilizar *templates* de composição no processo de recriação do conteúdo.

## 1.2. Motivação

Nas aplicações que se recriam dinamicamente, a lógica da aplicação é recriada - ou seja, uma especificação do documento NCL é removida e uma nova especificação é adicionada ao mesmo documento – o que ocorre a partir de um determinado evento. Em suma, o documento está sendo especificado novamente. Se a semântica da especificação adicionada é previamente conhecida por meio de um *template*, o gerador pode reusar essa especificação. Ao invés de recriar todos os objetos de mídia e relacionamentos da nova especificação do documento, só é preciso determinar quais são os componentes que preenchem as lacunas deixadas pelo *template*. O processo de recriação feito dessa forma se assemelha ao método de autoria orientado a *templates* descrito em (Soares Neto, 2010). Nesse método, o autor de aplicações tem seu trabalho simplificado porque ele só precisa especificar o documento de preenchimento, ao invés de especificar a semântica completa da aplicação. A Seção 2.2 apresenta maiores detalhes sobre esse método.

O desenvolvimento de aplicações NCL que se recriam dinamicamente com a utilização de *templates* é um cenário ainda não explorado em aplicações de TVD dinâmicas. Diante disso, a motivação desta dissertação consiste na possibilidade de disponibilizar um padrão de desenvolvimento a ser seguido por aplicações NCL dinâmicas que se enquadram na categoria das que se recriam dinamicamente. A ideia é utilizar um exemplo de aplicação NCL e, a partir da arquitetura dessa aplicação, fornecer uma base de conhecimento para que futuras aplicações desse mesmo tipo possam ser estruturadas segundo os princípios utilizados na aplicação exemplo.

## 1.3. Objetivos

Esta dissertação tem como principal objetivo especificar uma arquitetura baseada no uso de *templates* para geração de aplicações NCL recriadas dinamicamente. A partir dessa arquitetura, outras aplicações dinâmicas desse tipo podem se basear no mesmo processo e reaproveitar soluções recorrentes.

Como instrumentação para o alcance do objetivo principal, esta dissertação provê uma ferramenta de autoria gráfica para a geração da composição inicial da

aplicação em questão. Essa ferramenta também irá se valer dos princípios de autoria orientada a *templates* para facilitar o trabalho dos autores que não têm experiência com linguagens de programação.

#### **1.4. Organização da dissertação**

Na literatura, há abordagens que consistem em dar suporte à geração dinâmica de conteúdos e à especificação de aplicações com utilização de *templates*, o que é detalhado no Capítulo 2. De forma complementar, o Capítulo 3 apresenta diversos trabalhos relacionados à proposta desta dissertação.

O Capítulo 4 é dedicado à definição dos elementos arquiteturais presentes na solução de arquitetura proposta, bem como do papel de cada um desses elementos na estrutura de uma aplicação NCL que se recria dinamicamente. Algumas visões sobre a arquitetura dessa aplicação são apresentadas a fim de detalhar o processo de especificação da arquitetura em questão.

O Capítulo 5, por sua vez, apresenta a implementação de um suporte à geração automática de aplicações. Uma ferramenta gráfica permite ao autor gerar aplicações que seguem a arquitetura apresentada no Capítulo 4.

Por fim, o Capítulo 6 é reservado às considerações finais sobre o trabalho, destacando as principais contribuições desta dissertação e as sugestões de melhorias a serem realizadas em trabalhos futuros.