

Wizard para Autoria Gráfica de Documentos NCL com Templates

Douglas Paulo de Mattos, Júlia Varanda da Silva, Débora Christina Muchaluat Saade
Departamento de Ciência da Computação/Instituto de Computação, Laboratório MídiaCom
Universidade Federal Fluminense - UFF
R. Passo da Pátria, 156 – São Domingos – Niterói – RJ - 24210240
(21) 2629-5690
{douglas, julia, debora}@midia.com.uff.br

ABSTRACT

Using the declarative approach, interactive applications for the Brazilian digital TV system are developed using NCL (Nested Context Language). Although declarative languages facilitate building interactive TV applications, the author needs to have at least a basic knowledge of NCL. This paper proposes a wizard tool to allow the authoring of NCL documents for those with no knowledge about NCL. The proposed tool uses composite templates built with the XTemplate 3.0 language, which specify generic structures for NCL programs. The wizard presents a graphical interface which has drag-and-drop support for filling template components with specific media content nodes chosen by the author, making the creation of interactive TV applications easier. To do so, such media nodes are stored in a repository that organizes them by type. Furthermore, the graphical editing of content anchors for continuous media nodes is also provided. This wizard tool is a new version of a plugin of the graphical authoring environment for digital TV programs called NEXT (NCL Editor Supporting XTemplate).

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *graphical user interfaces (GUI), interaction styles*.
D.2.2 [Software Engineering]: Programming environments – *graphical environments, integrated environments*.

General Terms

Design, Languages.

Keywords

TV Digital Interativa, NCL, Ginga, XTemplate, template de composição hipermídia, wizard, ferramenta de autoria, drag and drop, editor gráfico.

1. INTRODUÇÃO

O padrão para o sistema de televisão digital interativa no Brasil utiliza a tecnologia de transmissão do padrão japonês e acrescenta inovações em algumas camadas do sistema de TV digital [1]. Na camada de middleware é apresentado o diferencial do padrão brasileiro, onde é especificado o GINGA [1], que pode utilizar linguagem declarativa ou procedural. No ambiente declarativo, utilizam-se as linguagens NCL (Nested Context Language) [7] e Lua [3] para o desenvolvimento de aplicações interativas, ambas desenvolvidas por universidades brasileiras.

A linguagem NCL é baseada no modelo conceitual hipermídia NCM (Nested Context Model) [8] que permite estruturar logicamente o documento através de contextos (conjuntos de nós e elos) e faz uso do paradigma baseado em eventos para relacionar os objetos de mídia temporalmente e espacialmente.

O modelo baseado em eventos é bastante expressivo para autoria de documentos multimídia interativos. Assim, a sincronização dos objetos de mídia (nós de contexto, vídeo, áudio, imagem, texto, programas Lua, etc.) é feita através de elos, cuja semântica da relação é definida por conectores hipermídia. Quando documentos NCL têm muitos componentes e relacionamentos entre eles, a definição de elos pode se tornar complicada para autores que não são especialistas em NCL.

Sendo assim, o propósito deste artigo é apresentar uma ferramenta do tipo wizard para facilitar a autoria de documentos NCL, ou seja, facilitando a criação de conteúdo interativo para o sistema brasileiro de TV digital, até mesmo para autores sem conhecimento da linguagem.

A fim de atender esta necessidade, a ferramenta apresentada neste artigo usa estruturas genéricas de contextos, chamadas de templates de composição, especificados com a linguagem XTemplate 3.0 [5], que podem ser reutilizados em diversos programas com conteúdo distinto, porém com uma estrutura de sincronização similar. Desta maneira, os relacionamentos de sincronização podem ser gerados automaticamente pelo módulo de autoria, sem necessitar que o autor conheça profundamente a linguagem.

A ferramenta, implementada em Java, possui uma interface bastante amigável permitindo aos autores construir suas aplicações através do recurso *drag-and-drop* (arrastar e largar) para preencher as telas dos templates que serão utilizados para criar a aplicação, de forma bem prática e simples. Para isso, a ferramenta oferece um repositório que armazena as mídias escolhidas pelo autor e as organiza de acordo com seu tipo em uma árvore de navegação, permitindo uma melhor visualização das mídias a serem utilizadas na aplicação NCL. Além disso, as âncoras de conteúdo (trechos de mídia) dos nós de mídia contínua também são definidas de forma intuitiva através do editor de âncoras oferecido pela ferramenta.

Esta ferramenta do tipo wizard é uma nova versão do plugin para utilização de templates de composição do editor gráfico de autoria para TV digital chamado NEXT [6], que oferece um conjunto de plugins para edição de um documento NCL.

A estrutura deste artigo é como se segue. Na Seção 2, são comentados alguns trabalhos relacionados. Na Seção 3, é apresentado o repositório de mídias, oferecido como um recurso auxiliar da ferramenta wizard. A Seção 4 apresenta a ferramenta wizard, detalhando as modificações realizadas em relação ao plugin original do NEXT. Finalmente, na Seção 5, as conclusões são apresentadas e os trabalhos futuros são apontados.

2. TRABALHOS RELACIONADOS

O trabalho desenvolvido por [9] apresenta um conjunto de editores gráficos de âncoras para objetos de mídia imagem, vídeo, áudio e texto. Esses editores tornam mais rápida e precisa a autoria de âncoras, principalmente em mídias contínuas que necessitam ser reproduzidas para melhor definir o intervalo de tempo desejado para cada âncora da mídia.

Outra ferramenta de autoria gráfica que também facilita o desenvolvimento de aplicações NCL é o Berimbau iTV [2], direcionada para profissionais que não possuem conhecimentos de programação e desejam desenvolver aplicações NCL para TV digital. A ferramenta possui uma interface prática e intuitiva que visa tornar a construção de documentos NCL bastante amigável para o autor e também possui um repositório de mídias para facilitar a busca das mídias que irão compor o documento. Entretanto, ela não oferece o uso de templates para auxiliar na construção das aplicações.

A ferramenta Composer 3 [4] é um ambiente de autoria integrado (IDE) que se adapta a diversos perfis de autores e com suporte a requisitos não funcionais. A arquitetura é baseada em um micronúcleo que permite a troca de informações entre os diferentes módulos que complementam o ambiente. Além do micronúcleo, um modelo central e extensões compõem a arquitetura. Suas extensões são construídas através de plugins, que são programas que interagem com o micronúcleo e acrescentam novas funcionalidades à ferramenta. Apesar do Composer facilitar e agilizar a criação de aplicações para TV digital, ele também não permite o uso de templates.

3. REPOSITÓRIO DE MÍDIAS

O repositório de mídias é uma ferramenta auxiliar que permite o armazenamento e visualização das mídias a serem utilizadas em uma aplicação NCL. As mídias são representadas graficamente no repositório como ícones para tornar mais fácil o processo de identificação do conteúdo desejado. Ou seja, para a mídia do tipo imagem, o ícone é representado pelo *thumbnail* (versão reduzida da imagem) da mídia; para mídia vídeo, o primeiro quadro do vídeo compõe o ícone; para os tipos de mídia áudio, texto e outros, são usados símbolos para representá-los. Na Figura 1, é ilustrada a interface gráfica do repositório de mídias.

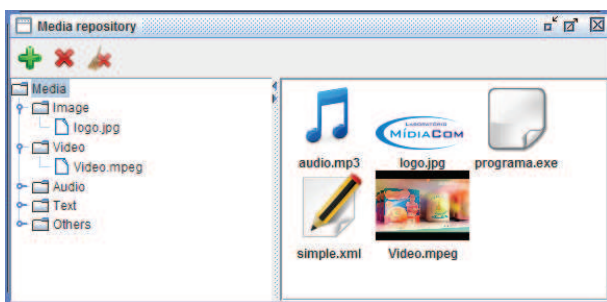


Figura 1. Interface gráfica do repositório de mídias.

Além disso, para facilitar a organização do conteúdo do repositório, é utilizada uma estrutura em árvore. A árvore de mídias possui cinco nós sendo que cada um armazena um tipo de mídia, ou seja, imagem, vídeo, áudio, texto e outros, e é apresentada do lado esquerdo na janela. Quando um dos nós é selecionado na árvore, são apresentadas as mídias que correspondem ao tipo desse nó. Também é possível exibir todas as mídias simultaneamente, selecionando-se o nó raiz (Media).

Existem três botões na ferramenta: o primeiro corresponde à inclusão de mídias no repositório, o segundo exclui a mídia selecionada e o último apaga todo o conteúdo do repositório. As mídias do repositório podem ser arrastadas para outras aplicações que suportam o recurso *drag-and-drop*. Este recurso é bastante utilizado na ferramenta proposta neste artigo.

As principais classes criadas para implementação do repositório são: *Repository*, *Tree*, *List*, *ButtonPanel*, *TransferList* e *Media*. A classe *Repository* é responsável por apresentar a árvore de mídias, a barra de botões e por exibir a lista de mídias. A classe *ButtonPanel* cria a barra de botões e implementa as ações destes. A classe *Tree* constrói a árvore de mídias e é responsável por mantê-la consistente conforme alterações feitas na lista de mídias. Ela também realiza a ação que deve ser tomada quando um de seus nós é selecionado. A *List* é responsável por guardar as mídias importadas e exibir as representações gráficas de cada tipo de mídia, as quais são criadas pela classe *Media*. A classe *TransferList* permite que as mídias do repositório sejam arrastadas para os plugins do editor gráfico NEXT. A Figura 2 mostra o diagrama de classes implementado.

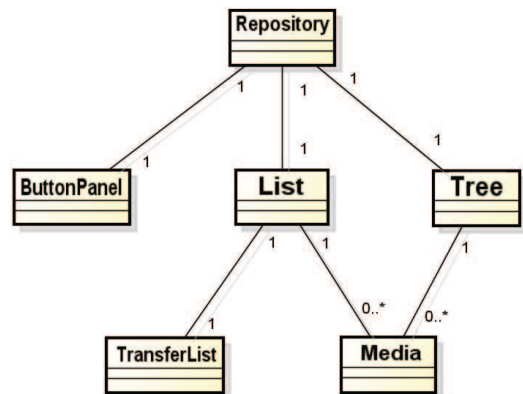


Figura 2. Diagrama de classes do repositório de mídias.

4. A FERRAMENTA WIZARD

O plugin para utilização de templates de composição do NEXT [6] facilita a autoria de aplicações para TV digital através do uso de templates especificados em XTemplate 3.0 [5]. Ele permite que autores possam construir suas aplicações em NCL indicando somente as mídias que irão compor o documento, sem precisar especificar a estrutura e os relacionamentos entre os nós do documento NCL. Além disso, ele agiliza a criação de aplicações que possuem um grande número de linhas de código e que muitas vezes são até repetitivas. Visando exemplificar essa última facilidade, considere um vídeo que possua 100 legendas. Para sincronizá-las com trechos do vídeo é necessária a criação de uma âncora para cada legenda e dos elos para relacioná-las. Assim, a digitação do código dessa aplicação se torna uma tarefa árdua e muito propícia a erros, porém utilizando o plugin de template, todo código é gerado automaticamente apenas com a definição das mídias (vídeo e legendas) pelo usuário.

O plugin utiliza um template como entrada e através da especificação do mesmo suas telas são geradas, indicando graficamente como os nós de mídia da aplicação serão apresentados simultaneamente na TV. A representação gráfica das telas do template indica o seguinte: os retângulos azuis representam os componentes da tela que devem ser preenchidos pelo autor, os vermelhos já são definidos pelo próprio template e as notas musicais representam um componente do tipo áudio que

pode ser azul ou vermelho, seguindo a mesma ideia dos retângulos. Esses componentes são preenchidos através de uma tabela de acordo com a tela atualmente selecionada, como ilustrado na Figura 3.



Figura 3. Interface gráfica do plugin original.

Em [6] foram realizados testes de usabilidade com 10 alunos de computação que não conheciam a linguagem NCL e dois alunos que já a conheciam, a fim de avaliar a funcionalidade e praticidade do plugin de templates. Após os testes, os mesmos deram sugestões para o aprimoramento do plugin. As principais sugestões dadas foram: arrastar mídias de um repositório diretamente para a tela do template, exibir a imagem da mídia escolhida no lugar dos retângulos e melhorar a indicação de quando uma âncora pode ser criada na mídia.

A ferramenta wizard, proposta neste artigo, visa facilitar o preenchimento dos componentes de um template oferecendo praticidade e rapidez. Assim, tal preenchimento deixou de ser feito através de tabelas e passou a ser feito na própria tela do template. Ou seja, quando o autor seleciona uma das telas do template na lista de telas, no lugar da tabela é exibida a própria tela selecionada, em tamanho maior, para que seja possível o preenchimento através do recurso *drag-and-drop*. Para isto, foi usado como ferramenta auxiliar, o repositório de mídias. A Figura 4 mostra como o repositório e a ferramenta wizard são usados em conjunto.

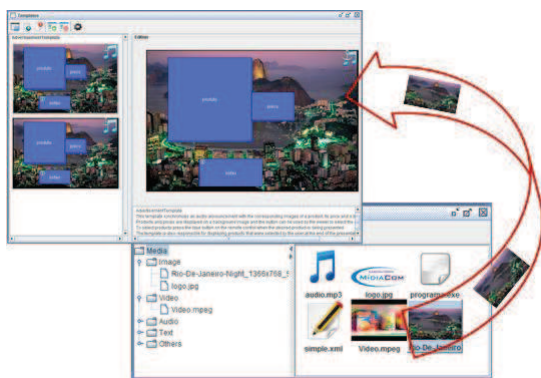


Figura 4. Drag and Drop.

Quando o autor deseja preencher algum componente da tela de um template, ele abre o repositório, seleciona a mídia desejada e a arrasta para dentro do componente a ser preenchido na tela do template. Para um componente (retângulo azul) que receba alguma mídia imagem, ele passa a exibir a imagem desta mídia; caso seja um vídeo, ele exibe o primeiro quadro do vídeo; caso a mídia seja do tipo texto, o retângulo passa a ser de cor verde e o conteúdo textual é exibido no centro do retângulo; e para o tipo outros, é exibido um ícone indicando seu preenchimento. No preenchimento de componente áudio, o símbolo que o representa é alterado para cor verde. A Figura 5 mostra como

cada tipo de componente da tela do template é modificado após a escolha da mídia. Os componentes que definem um nó NCL específico, já pré-definido no template, e cujo conteúdo não poderá ser redefinido, deixam de ser representados por um retângulo vermelho e passam, nesta nova versão, a exibir seu conteúdo conforme regras já ditas.

Preenchimento		Tipode Mídia
		Imagem
		Áudio
		Texto
		Outros
		Vídeo

Figura 5. Alteração dos componentes após o preenchimento

Para melhor explicar a funcionalidade específica de um template, a ele é associado um arquivo RDF (*Resource Description Framework*) explicativo. Na primeira versão do plugin, somente a funcionalidade geral do template era exibida ao usuário. No wizard, a partir do documento RDF, além de exibir a explicação geral da função do template, para cada componente da tela do template, também é exibida sua descrição quando o cursor do mouse é posicionado sobre o mesmo, facilitando ainda mais o entendimento do template pelo usuário do wizard.

A criação de âncoras em um componente, na primeira versão do plugin, é feita clicando-se no botão que representa a âncora, localizado na tabela de componentes (vide Figura 3). Porém, este botão está presente em todos os componentes, mesmo quando estes não podem ter âncoras, o que pode confundir o autor. Deste modo, na nova versão, o ícone de âncora é apresentado sobre o desenho somente dos componentes que podem ter âncoras. E, para sua criação ou edição, basta clicar sobre o ícone. A Figura 6 apresenta esta nova indicação de âncora.

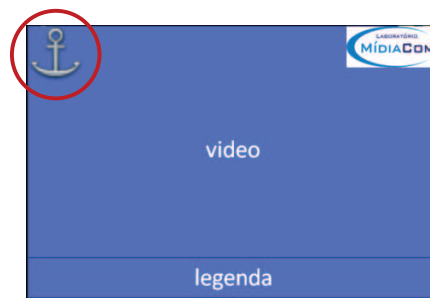


Figura 6. Representação de âncora em um componente

A janela de criação e edição de âncoras de mídias contínuas possui um player que foi implementado utilizando a biblioteca JMF (Java Media Framework) para reprodução da mídia áudio ou vídeo. Assim, é possível determinar o trecho (início e fim) que define uma âncora da mídia usando a linha do tempo de reprodução da mídia no player. Essa mesma tela exibe ainda a lista de âncoras já criadas para esse componente. Nessa nova versão, a edição de âncoras se tornou mais intuitiva e prática. A Figura 7 mostra a nova interface da janela de criação de âncoras.

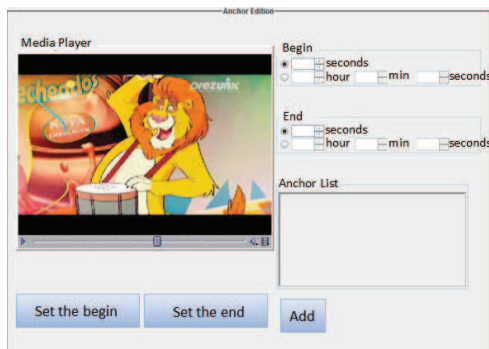


Figura 7. Tela de edição de âncoras.

A implementação desta ferramenta se dá com as seguintes classes principais: *StartPlugin*, *PluginInfo*, *MyLayeredPane*, *MyList*, *Screen*, *Template*, *TempScreen*, *VocabularyComp*, *Picture*, *MediaPanel* e *PlayerEventHandler*. A classe *StartPlugin* é responsável por iniciar o plugin que será chamado pelo NEXT quando o autor desejar utilizá-lo. *PluginInfo* fornece ao NEXT informações sobre o plugin, como o seu apelido e sua lista de elementos NCL de interesse. A classe *MyLayeredPane* tem como função armazenar todas as telas do template, que são apresentadas pela *MyList*, em um panel especial, o qual permite apresentar a tela selecionada na lista em tamanho maior. A classe *Template* armazena as informações do template de composição e a ela está associada a classe *TempScreen*, que armazena todas as telas do template. Já a classe *Screen* corresponde a apenas uma tela do template. As telas são guardadas pela *MyList* e *MyLayeredPane*.

Para armazenar as informações de cada componente do template, foi criada a classe *VocabularyComp*, a qual permite saber o tipo de conteúdo do componente, se este possui interface e se ele é editável ou se foi definido pelo próprio template. Cada componente tem uma representação gráfica associada que será usada para desenhar as telas do template, tal representação é dada pela classe *Picture*. Já as classes *MediaPanel* e *PlayerEventHandler* trabalham juntas para permitir que se edite ou crie âncoras através do player oferecido pela API JMF. O diagrama de classes é apresentado na Figura 8.

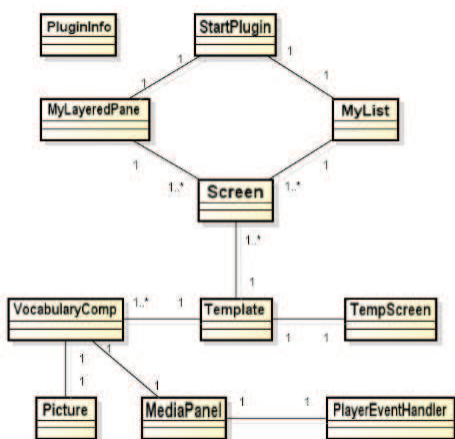


Figura 8. Diagrama de classe da ferramenta wizard.

5. CONCLUSÃO

Este trabalho teve como principal objetivo a construção de uma ferramenta wizard, como uma nova versão do plugin de

templates do NEXT, para melhorar o preenchimento das telas dos templates de composição, tornando-o mais prático e rápido. Para isso foi utilizado o recurso *drag-and-drop* para preencher as telas do template, que são exibidas em tamanho maior, cujos componentes apresentam o conteúdo da mídia arrastada ao invés de utilizar uma tabela de componentes. A indicação de âncora passou a ser bem mais clara, visto que somente componentes que podem ter âncoras apresentam o ícone da mesma sobre sua representação. E ainda, para tornar mais precisa e rápida a edição de âncoras, utilizou-se a API JMF, permitindo a reprodução da mídia, áudio ou vídeo, para determinar o intervalo de tempo da âncora do componente.

Como trabalho futuro, será desenvolvido um novo plugin para o NEXT, oferecendo a visão temporal de documentos NCL e permitindo a edição da cadeia temporal da aplicação. Ele ainda terá uma estratégia para tratar a interatividade das aplicações, visto que isso torna a exibição temporal uma tarefa não trivial. O objetivo é tornar mais fácil a alteração da ordem de apresentação das mídias de uma aplicação NCL através de uma interface gráfica amigável, para que autores sem grande conhecimento em NCL possam editar a ordem temporal das mídias em suas aplicações.

6. AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelo CNPq, CAPES e FAPERJ.

7. REFERÊNCIAS

- [1] ABNT NBR 15606-2:2007. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 2: Gínga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Setembro 2007.
- [2] Berimbau iTV Author, <http://www.batuque.tv/>, acessado em novembro de 2011.
- [3] Jerusalimschy, R.; Figueiredo, L.H.; Celes, Waldemar. "Lua 5.1 Reference Manual." Lua.org, Agosto de 2006. ISBN 85-903798-3-3.
- [4] Lima, B. S., Azevedo, R. G., Moreno, M. F. e Soares, L. F. – Composer 3: Ambiente de autoria extensível, adaptável e multiplataforma. WebMedia – Workshop de TV Digital Interativa (WTVDI), 2010.
- [5] Santos, J. A. F. and Muchalua Saade, D. C. Linguagem XTemplate 3.0: Facilitando a Autoria de Programas NCL para TV Digital Interativa. In Anais do XV Simpósio Brasileiro de Sistemas Multimídia e Web, 2009.
- [6] Silva, J.V e Muchalua Saade, D. C. NEXT – Editor Gráfico para Autoria de Documentos NCL com Suporte a Templates de Composição, In Anais do XVIII Simpósio Brasileiro de Sistemas Multimídia e Web, 2012.
- [7] Soares, L.F.G.; Barbosa, S.D.J. Programando em NCL 3.0 – Desenvolvimento de Aplicações para o Middleware Gínga, TV Digital e Web. Elsevier. 2009.
- [8] Soares, L.F.G., Rodrigues, R.F. Nested context model 3.0 part 1 – ncm core. Technical report, PUC-Rio, Rio de Janeiro, 2005.
- [9] Vernot, Lucas ; Amorim, T.; Muchalua Saade, D. C. Editores de Âncoras para Objetos Multimídia. In: Simpósio Brasileiro de Sistemas Multimídia e Web/Workshop de Iniciação Científica, 2005.