

Editor Gráfico de Conectores Hipermedia para Linguagem NCL 3.0*

Júlia Varanda da Silva, Débora Christina Muchaluat-Saade
Laboratório MídiaCom, Departamento de Ciência da Computação
Universidade Federal Fluminense

R. Passo da Pátria, 156 – Bloco E – Sala 408 – São Domingos, Niterói, RJ, Brasil

jvaranda@ic.uff.br, debora@midia.com.uff.br

ABSTRACT

Significant development effort is dedicated to authoring spatio-temporal relationships, including user interactivity, in multimedia applications. That effort is even higher when the language requires specification of relationships using links, as it happens with NCL, adopted in the Brazilian digital TV system. In an NCL program, the author must specify each type of relation to be used, as a causal hypermedia connector, and each relationship among media components as a link that uses a connector, which makes the authoring process very tiring. Connector specification requires knowledge about the event-based temporal model used by NCL, which is not trivial for beginners. This paper presents a graphical editor for NCL hypermedia connectors. Besides facilitating connector creation, the proposed editor can help users understanding how spatio-temporal relations are specified in an interactive digital TV program.

RESUMO

Grande parte do esforço de desenvolvimento de uma aplicação multimídia é dedicada à especificação de relacionamentos espaço-temporais, incluindo a possibilidade de interatividade do usuário. Esse esforço é ainda maior quando a linguagem de autoria requer a especificação de relacionamentos através de elos, como é o caso da linguagem NCL, adotada no Sistema Brasileiro de TV Digital. Em um programa NCL, o autor deve especificar cada tipo de relação a ser usada, como um conector hipermedia causal, e cada relacionamento entre os componentes de mídia como um elo que utiliza um dos conectores especificados, tornando a autoria trabalhosa. A especificação de conectores requer domínio do modelo temporal baseado em eventos que NCL utiliza, o que não é trivial para iniciantes na linguagem. Este artigo apresenta um editor gráfico de conectores hipermedia. Além de facilitar a criação de conectores, o editor gráfico pode ajudar autores com pouco conhecimento de NCL a entender melhor como as relações espaço-temporais são especificadas em um programa interativo para TV digital.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *graphical user interfaces (GUI), interaction styles.*

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Short Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9650.
SBC - Brazilian Computer Society

*Este trabalho foi parcialmente financiado pela CAPES, CNPq e FAPERJ.

General Terms

Design, Languages.

Keywords

NCL language, digital TV, hypermedia connectors, graphic editor, spatio-temporal relations, interactivity.

1. INTRODUÇÃO

Grande parte do esforço de desenvolvimento de uma aplicação multimídia é dedicada à especificação de relacionamentos espaço-temporais, incluindo a possibilidade de interatividade do usuário. Esse esforço é ainda maior quando a linguagem de autoria declarativa requer a especificação desses relacionamentos através de elos e conectores, como é o caso da linguagem NCL (*Nested Context Language*) [1], adotada no Sistema Brasileiro de TV Digital (SBTVD) através do middleware Ginga-NCL.

Com o objetivo de minimizar as dificuldades relacionadas à criação de aplicativos, diferentes sistemas de autoria multimídia são propostos, oferecendo ferramentas gráficas que facilitam o trabalho do autor, escondendo detalhes de sintaxe da linguagem utilizada e muitas vezes ajudando-o a compreender o processo de autoria. O propósito deste artigo é apresentar uma ferramenta gráfica que foi produzida com o intuito de facilitar a criação de conectores hipermedia causais, os quais são elementos indispensáveis em um documento multimídia que utiliza a linguagem NCL.

O restante do texto apresenta a seguinte estrutura. Na Seção 2, são apresentados alguns trabalhos relacionados. Na Seção 3, são apresentadas algumas características da linguagem NCL com foco na especificação de conectores. A Seção 4 explica o funcionamento do editor gráfico de conectores e apresenta o resultado do teste de usabilidade que foi realizado. Finalmente, na Seção 5, são apresentadas as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

DEMAIS [2] é uma ferramenta de autoria que oferece um ambiente para a criação de protótipos de conteúdo. A ferramenta é composta por um conjunto de storyboards, permitindo adicionar sincronizações entre os elementos deste, e/ou entre outros storyboards, definir comportamentos baseados no tempo ou em eventos, entre outros. Para permitir a especificação de um comportamento através de uma anotação, DEMAIS utiliza um parser baseado na gramática LL 1, a qual suporta um conjunto de sentenças, no entanto, para isto é preciso que o autor conheça as regras de sintaxe da gramática. Para editar eventos de origem e de destino, utiliza-se uma linguagem visual baseada em ícones. Cada ícone de evento expressa uma condição que, ao ocorrer no elemento de origem, causará uma ação no elemento de destino.

Para alterar um evento, basta clicar no mesmo e, depois, selecionar o evento a ser alterado.

O Composer [5] é um ambiente de autoria voltado para a criação de programas que utilizam a linguagem NCL. Esta ferramenta oferece, além da visão textual, as visões gráficas estrutural, de layout e temporal, as quais funcionam sincronamente, oferecendo um ambiente integrado. Modificações no documento podem ser feitas a partir de qualquer uma das visões oferecidas. Sua interface gráfica é composta por uma barra de menus e outra de ferramentas, as quais oferecem funcionalidades de acordo com a visão que está sendo utilizada pelo autor. No entanto, o Composer não oferece uma interface gráfica para a criação de conectores.

EDITEC [3] é um editor gráfico de templates de composição multimídia baseado na linguagem XTemplate 3.0. O sistema oferece três visões permitindo ao usuário ter uma melhor visualização do template durante o processo de autoria. A visão estrutural é a principal visão da ferramenta e nela o usuário pode criar os elementos e definir os relacionamentos que farão parte dos templates, de acordo com a linguagem XTemplate. Ela é dividida em duas abas, Vocabulary View e Body View, a primeira apresenta os elementos da linguagem e a segunda, apresenta os relacionamentos espaço-temporais entre os componentes do template. Apesar de conectores serem utilizados na definição de templates de composição, EDITEC não oferece um editor de conectores.

O objetivo do trabalho apresentado em [6] foi propor uma criação visual para as relações espaço-temporais. Uma das principais preocupações foi utilizar figuras intuitivas para o usuário, como as utilizadas em um controle remoto, por exemplo. Para validar seu propósito de como a abordagem visual é poderosa, uma representação visual foi mapeada para o código NCL e um protótipo foi desenvolvido e incorporado ao Composer. Além de utilizar ícones para representar estas relações, uma notação textual também é utilizada para oferecer um feedback ao autor. Apesar de apresentar uma representação visual das relações, não foi possível identificar no trabalho como o autor irá criar as relações através dessa representação.

3. CONECTORES HIPERMÍDIA NCL

A linguagem NCL versão 3.0 é uma linguagem declarativa para autoria de documentos multimídia baseada no padrão XML [4] e no modelo conceitual NCM. NCM utiliza o paradigma temporal baseado em eventos [9] para especificar o sincronismo espaço-temporal e a interatividade em um programa de TV digital.

O sincronismo em um documento NCL é expresso através de elos, que representam relacionamentos causais entre os componentes da aplicação, utilizando conectores hipermídia [7], os quais definem os papéis dos participantes da relação de sincronismo.

A definição de um conector é feita através da máquina de estados de eventos, ilustrada na Figura 1. Os tipos de eventos, que podem ser usados na definição de um conector, são apresentação, seleção e atribuição. Um conector causal define ações que devem ser executadas sobre eventos, quando determinada condição sobre eventos for satisfeita.

Um elo pode interligar diversas mídias e cada uma destas é associada a um papel do conector utilizado pelo elo. Papéis podem ser de condição ou de ação e podem ser simples ou compostos. Um papel de condição simples especifica uma condição sobre um determinado evento e, um papel de ação

simples, especifica qual ação um evento sofrerá. Ações provocam transições nos estados dos eventos.

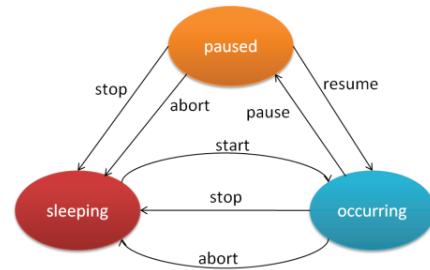


Figura 1. Máquina de estado de eventos.

Uma condição composta é formada por duas ou mais condições simples ou compostas, e é representada por uma expressão lógica com os operadores e (AND) e ou (OR). Uma ação composta é formada por duas ou mais ações simples ou compostas, que devem ser executadas em paralelo, sendo disparadas em uma ordem explícita (SEQ) ou em qualquer ordem (PAR).

Ao criarmos um papel, seja de condição ou de ação, alguns atributos precisam ser definidos. Os obrigatórios são: *role*, *eventType* e *transition*, para condição, ou *eventAction*, para ação. Estes representam, respectivamente, o identificador do papel (que deve ser único), o tipo de evento associado e a transição/ação na máquina de estados. Os outros atributos, não obrigatórios, permitem uma maior expressividade ao conector. Por conveniência, alguns papéis predefinidos são oferecidos pela linguagem. Estes já possuem os atributos obrigatórios definidos e são reconhecidos pelo formatador NCL através de nomes (identificadores) reservados.

Além dos papéis, um conector também permite a definição de parâmetros, os quais oferecem uma melhor reutilização do conector em elos distintos. Com tantos atributos e possibilidades, conectores podem ser bastante complexos, principalmente quando usamos condições e ações compostas. O conector da Figura 2, retirado de [8], é um exemplo.

```
<causalConnector id='onEndAttrSetTrueResume'>
  <compoundCondition operator='and'>
    <simpleCondition role='onEndAttribution'>
      <assessmentStatement comparator='eq'>
        <attributeAssessment role='testIfFalse'
          eventType='attribution'
          attributeType='nodeProperty'>
          <valueAssessment value='false'>
        </assessmentStatement>
      </compoundCondition>
    <compoundAction operator='seq'>
      <simpleAction role='resume'>
      <simpleAction role='setAsTrue' value='true'
        eventType='attribution' actionType='start'>
    </compoundAction>
  </causalConnector>
```

Figura 2. Código NCL de um conector mais elaborado.

4. EDITOR GRÁFICO DE CONECTORES

Desenvolvido em Java para oferecer portabilidade em diferentes plataformas, o editor de conectores apresenta uma interface gráfica simples e exibe mensagens para orientar o usuário durante a criação do conector. A Figura 3 mostra a interface principal do editor exibindo o conector apresentado na Figura 2.

Na Figura 3, a região A exibe uma árvore contendo todos os conectores contidos na base de conectores em uso. Qualquer um desses conectores pode ser editado e novos conectores criados são inseridos na árvore.

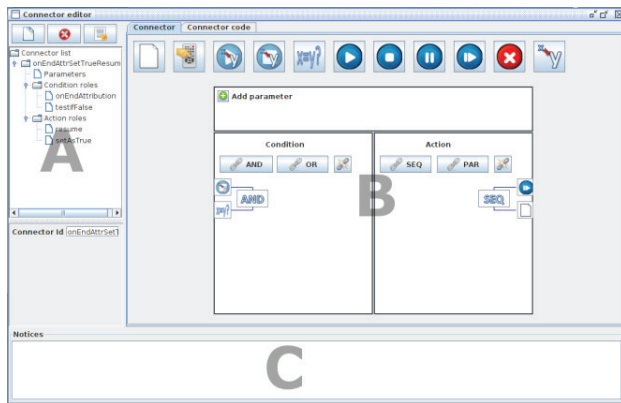


Figura 3. Editor gráfico de conectores

Ao selecionar um conector qualquer da árvore, o mesmo é exibido na região B, a qual possui duas abas. A primeira exibe uma representação gráfica do conector e, a segunda, o código NCL do mesmo. Na parte superior, são apresentados onze ícones, os quais representam diferentes tipos de papéis simples. O primeiro permite que o usuário crie um papel e defina todos os seus parâmetros. Com exceção do quinto ícone, que representa um papel de *assessmentStatement*, os outros ícones representam os papéis pré-definidos pela linguagem NCL. A Tabela 1 mostra estes ícones.

Tabela 1. Ícones dos papéis de ação e condição.

Ícone	Papel de condição	Papel de ação	Significado
	definido pelo usuário	definido pelo usuário	Permite criar um novo papel
	onSelection	-	Seleção do usuário (interatividade)
	onBeginAttribution	-	No início da atribuição
	onEndAttribution	-	No fim da atribuição
	definido pelo usuário	-	Avalia o estado de um evento ou o valor de um atributo
	onBegin	start	Início/Iniciar apresentação
	onEnd	stop	Fim/Finalizar apresentação
	onPause	pause	Pausa/Pausar apresentação
	onResume	resume	Retomar apresentação
	onAbort	abort	Abortar apresentação

	-	set	Atribuir novo valor a um atributo
--	---	-----	-----------------------------------

Os ícones de papéis servem tanto para criar papéis de condição como de ação, basta que o papel desejado seja arrastado até o desenho do conector, no espaço destinado a papéis de condição (lado esquerdo) ou de ação (lado direito). Ao soltar o ícone arrastado no conector, será aberta uma janela exibindo os atributos correspondentes ao papel, os quais podem ser editados pelo usuário. Na Figura 4, a janela de criação de um papel de ação é apresentada.

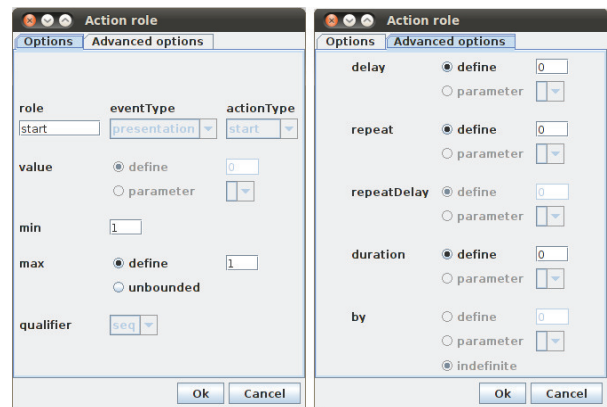


Figura 4. Janela de criação de um papel de ação.

O papel de condição do tipo *assessmentStatement*, que representa um tipo especial de papel de condição, é criado da mesma forma que os outros, sua diferença está nos atributos apresentados na janela de criação/edição, já que estes não são iguais aos dos outros papéis.

A ferramenta também permite a criação de condições/ações compostas. Cada um dos lados do conector possui ícones para tal criação. Depois de adicionar os papéis simples, basta que o usuário selecione os papéis desejados e clique no operador adequado. Os próprios ícones indicam os operadores a serem utilizados na criação da condição/ação composta. Para remover algum papel de uma condição/ação composta, basta selecionar o papel e utilizar o botão “Desagrupar”. Os ícones utilizados para criar condições/ações compostas, assim como o botão “Desagrupar”, podem ser visualizados na Tabela 2.

Tabela 2. Ícones utilizados para criação/edição de condições e ações compostas.

Ícone	Operador	Significado
	AND	Todas as condições devem ser satisfeitas.
	OR	Pelo menos uma das condições deve ser satisfeita.
	SEQ	Ações devem ser executadas em ordem explícita.
	PAR	Ações devem ser executadas em ordem aleatória.
	-	Desagrupar papel de condição ou ação composta.

Na parte superior do desenho do conector, existe um botão que permite a criação de parâmetros. Depois de criado, o parâmetro será exibido juntamente com um botão para removê-lo, caso o usuário não queira mais utilizá-lo.

A parte inferior da Figura 3, região C, é destinada a mensagens de ajuda para auxiliar na criação do conector. Enquanto esta região apresentar alguma mensagem, significa que o conector ainda não está terminado, ou seja, não está pronto para ser utilizado em um documento NCL.

4.1 Teste de usabilidade do editor

A fim de verificar a usabilidade do editor proposto, este foi disponibilizado para 17 estudantes já com algum conhecimento da linguagem NCL, alguns mais experientes, outros menos. O teste consistiu em uma lista de conectores que os estudantes deveriam criar utilizando o editor gráfico. Ao final, eles responderam um formulário, onde opiniões sobre os ícones utilizados na interface para a representação dos papéis e sugestões foram solicitadas.

O formulário era composto por perguntas que variavam desde o grau de conhecimento da linguagem NCL à importância da ferramenta proposta. As questões eram objetivas e as opções de resposta variavam numa escala de seis opções (de zero a cinco, onde zero era a pior opção e cinco a melhor). A Tabela 3 apresenta o resultado obtido.

Tabela 3. Perguntas e média dos resultados obtidos no questionário sobre usabilidade da ferramenta.

Pergunta	Nota Média	Conclusão do resultado
Como você classifica o seu conhecimento em NCL?	3	Bom
Como você classifica o seu conhecimento em conectores NCL?	3	Bom
Você já criou algum aplicativo utilizando a linguagem NCL?	-	Sim
Com que frequência você cria os conectores que utiliza nas suas aplicações?	3	De vez em quando
Com relação ao tempo para compreender a ferramenta, que nota você daria à mesma?	4	Rápido
Com relação ao grau de usabilidade (facilidade de uso), que nota você daria à ferramenta?	4	Muito bom
Que nota você daria para a importância (necessidade) da ferramenta?	4	Muito útil
Com relação ao grau de desempenho (tempo de processamento, falhas do programa), que nota você daria à ferramenta?	4	Muito bom
Com relação aos ícones utilizados para representar os papéis, que nota você daria em relação à clareza dos mesmos? Alguma sugestão?	3	Claros
Você conseguiu criar os conectores da lista?	-	Sim, todos

Analisando-se as respostas, conclui-se que a maioria dos alunos levou pouco tempo para compreender sobre como utilizar a ferramenta e que os graus de usabilidade e de desempenho (falhas da ferramenta, tempo de processamento) foram muito bons. A maioria dos alunos considerou os ícones utilizados para

representar os papéis como “claros”. Apesar de a maioria dos alunos nem sempre criarem os conectores que utilizam em seus documentos NCL, estes consideraram a ferramenta muito útil.

5. CONCLUSÃO

A principal contribuição deste artigo foi a apresentação de uma ferramenta gráfica para a criação de conectores hipermídia na linguagem NCL. O trabalho visa facilitar a criação de conectores, que são elementos importantes utilizados para especificar o sincronismo em aplicações NCL. Testes de usabilidade também foram realizados para verificar a qualidade do trabalho e adquirir sugestões de melhoria.

O editor gráfico de conectores apresentado pode ser usado como uma aplicação Java standalone ou como módulo (plugin) de um editor gráfico mais completo, como por exemplo um editor NCL, como o Composer, ou um editor de templates de composição, como o EDITEC.

Como trabalho futuro, algumas funcionalidades podem ser adicionadas à ferramenta, algumas sugeridas pelos alunos foram a utilização de um menu para a criação de papéis e uma forma alternativa para a remoção dos mesmos.

6. REFERÊNCIAS

- [1] ABNT NBR 15606-2:2007. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 2: Giga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Setembro 2007.
- [2] Bailey, B. P., Kostan, J. A., Carlis, J. V. – DEMAIS: Designing Multimedia Applications with Interactive Storyboards. 9th ACM International Conference on Multimedia, páginas 241-250, Ottawa, 2001.
- [3] Damasceno J. R., Santos J. A. F e Muchalut-Saade D. C., EDITEC: Editor Gráfico de Templates de Composição para Facilitar a Autoria de Programas para TV Digital Interativa. WebMedia, Belo Horizonte, 2010.
- [4] Extensible Markup Language (XML) 1.1 (Second Edition) - W3C Recommendation 16 August 2006, Setembro 2006, <http://www.w3.org/TR/2006/REC-xml11-20060816/>, acessado em abril de 2011.
- [5] Guimarães R. L., Costa R. M. R. e Soares L. F. G. Composer: Ambiente de Autoria de Aplicações Declarativas para TV Digital Interativa. WebMedia, Gramado, outubro de 2007.
- [6] Guimarães R. L., Neto C. S. S. e Soares L. F. G. – A Visual Approach for Modeling Spatiotemporal Relations. DocEng '08 Proceeding of the eighth ACM Symposium on Document Engineering, páginas 285-288, Nova Iorque, 2008.
- [7] Muchalut-Saade D. C. e Soares L. F. G., XConnector e XTemplate: Estendendo XLink para Aumentar Expressividade e Reuso, VIII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia - SBMídia2002, Fortaleza, Outubro de 2002
- [8] Soares L. F. G e Barbosa S. D. J. Programando em NCL 3.0. Editora Campus/Elsevier, 2009.
- [9] Soares L. F. G. e Rodrigues R. F. Nested Context Model 3.0 Part 1 - NCM Core. Monografia em Ciência da Computação, Departamento de Informática, PUC-Rio, Rio de Janeiro, Maio 2005.