

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Heron Vilela de Oliveira e Silva

**X-SMIL: Aumentando Reuso e
Expressividade em Linguagens de
Autoria Hipermídia**

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA

Programa de Pós-Graduação em Informática

Rio de Janeiro

Abril de 2005

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Heron Vilela de Oliveira e Silva

**X-SMIL: Aumentando Reuso e Expressividade em
Linguagens de Autoria Hipermedia**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientadore: Luiz Fernando Gomes Soares
Co-orientador: Rogério Ferreira Rodrigues



Heron Vilela de Oliveira e Silva

X-SMIL: Aumentando Reuso e Expressividade em Linguagens de Autoria Hipermedia

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Luiz Fernando Gomes Soares

Orientador

Departamento de Informática - PUC-Rio

Rogério Ferreira Rodrigues

Co-orientador

Departamento de Informática - PUC-Rio

Marco Antonio Casanova

Departamento de Informática - PUC-Rio

Renato Fontoura de Gusmão Cerqueira

Departamento de Informática - PUC-Rio

José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 4 de abril de 2005

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Heron Vilela de Oliveira e Silva

Formou-se em Engenharia de Computação pela PUC-Rio em 2002. Atualmente, integra o grupo de pesquisadores do Laboratório TeleMídia da PUC-Rio, desenvolvendo pesquisa na área de Sistemas Hipermídia.

Ficha Catalográfica

Silva, Heron Vilela de Oliveira e

X-SMIL: aumentando reuso e expressividade em linguagens de autoria hipermídia / Heron Vilela de Oliveira e Silva ; orientador: Luiz Fernando Gomes Soares. – Rio de Janeiro : PUC-Rio, Departamento de Informática, 2005.

210 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática .

Inclui referências bibliográficas

1. Informática – Teses. 2. Sistemas hipermídia. 3. Autoria. 4. Linguagens declarativas. 5. Relações. 6. SMIL. 7. Sincronização. 8. Conectores. 9. Templates. 10. NCL. I. Soares, Luiz Fernando Gomes. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Este trabalho é dedicado:

À Deus, por ser.

Agradecimentos

À minha família, pelo amor incondicional.

Aos meus orientadores, pela amizade, compreensão e apoio constante.

Aos amigos, por estarem, mesmo quando ausentes, sempre presentes.

Ao espírito TeleMídia.

Ao DI.

À PUC, CAPES, CNPq e FUNTTEL.

Resumo

Silva, Heron Vilela de Oliveira. **X-SMIL: Aumentando Reuso e Expressividade em Linguagens de Autoria Hipermídia.** Rio de Janeiro, 2005. xxxp. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho está inserido no contexto de ambientes de autoria e execução hipermídia, sendo as linguagens declarativas para autoria de documentos o seu foco principal. Tendo-se como objetivo aumentar a expressividade e o reuso na especificação de documentos hipermídia, este trabalho introduz as linguagens X-SMIL e NCL - Nested Context Language - versão 2.1. Utilizando-se o conceito de templates, X-SMIL permite a definição de novas semânticas para composições SMIL, além dos tradicionais elementos seq, par e excl. Templates, em X-SMIL, são especificados em um perfil de XTemplate, que estende a idéia original da linguagem XTemplate de NCL. Com base nas novas facilidades para definição de templates, esse perfil foi usado para especificar a linguagem NCL 2.1. X-SMIL também permite a especificação de conectores hipermídia, tratando relações hipermídia como entidades de primeira classe - funcionalidade incorporada em X-SMIL pelo uso do módulo XConnector de NCL. Outro objetivo deste trabalho é o de apresentar um framework para o processamento de documentos XML. Utilizando-se esse framework, diversos compiladores foram implementados, o que possibilitou, entre outras funcionalidades, a conversão de documentos NCL em especificações SMIL ou X-SMIL e vice-versa.

Palavras-chave

sistemas hipermídia; autoria; linguagens declarativas; relações; elos; sincronização; conectores; templates; NCL; SMIL; XML; compilador

Abstract

Silva, Heron Vilela de Oliveira. **X-SMIL: Improving Reuse and Expressiveness in Hypermedia Authoring Languages**. Rio de Janeiro, 2005. XXXp. Master Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This work is related to hypermedia authoring and execution environments, and its main focus is declarative document authoring. Aiming at improving the expressiveness and reuse in the specification of hypermedia documents, this work introduces the hypermedia authoring languages X-SMIL and NCL - Nested Context Language - version 2.1. Exploiting the concept of templates, X-SMIL allows the definition of new semantics for SMIL compositions, besides its usual seq, par and excl elements. X-SMIL templates are specified using an XTemplate profile, which extends the original idea of the NCL XTemplate language. Bringing new facilities for template definitions, this new profile is used to further improve the NCL language. X-SMIL also offers support for handling hypermedia relations as first-class entities, through the use of hypermedia connectors - brought to X-SMIL via the NCL XConnector module. Another important goal of this work is to present a framework to facilitate the development of XML documents parsing and processing tools. Based on this framework, several compilers were implemented, permitting, among other features, the conversion of NCL documents into SMIL or X-SMIL specifications and vice-versa.

Palavras-chave

hypermedia systems; authoring; declarative languages; relations; links; synchronization; connectors; templates; NCL; SMIL; XML; compiler

Sumário

1	Introdução	15
1.1.	Motivação	15
1.2.	Objetivos	19
1.3.	Organização da Dissertação	22
2	Linguagens para Descrição de Documentos Hipermedia	23
2.1.	SGML e XML	24
2.2.	HTML e XHTML	26
2.3.	SMIL	27
2.4.	NCL	29
3	<i>Nested Context Language 2.1</i>	34
3.1.	Funções de Custo	34
3.2.	Regras de Apresentação	36
3.3.	Refinamentos de NCL 2.1	40
3.4.	Linguagem XConnector	46
3.5.	Linguagem XTemplate	51
4	X-SMIL	64
4.1.	XT-SMIL: SMIL + XTemplate	64
4.2.	SMIL + XConnector (XC-SMIL) e X-SMIL	70
5	<i>Framework</i> para Compiladores	74
5.1.	<i>Framework</i> Genérico para Processamento	76
5.2.	<i>Framework</i> para Compiladores de Documentos NCL	85
5.2.1.	Funcionamento do <i>Framework</i> de Compiladores NCL	86
5.2.2.	Compiladores de documentos NCL	89
5.3.	<i>Framework</i> para Compiladores de Documentos SMIL	93
5.4.	X-SMIL	95

5.5. Compiladores XConnector e XTemplate	96
6 Trabalhos Relacionados	107
6.1. <i>Template</i> de Composição	107
6.2. Processamento de documentos XML	110
6.3. <i>Framework</i> e Compiladores	112
6.4. Compiladores XML	117
6.5. Extensões à SMIL	118
6.6. Conversão entre Modelos	120
7 Conclusão e Trabalhos Futuros	123
7.1. Templates	124
7.2. Conversões entre Formatos	127
8 Referências Bibliográficas	130
9 Apêndice A	136
9.1. NCL 2.0	136
9.2. NCL 2.1	138
10 Apêndice B	142
10.1. NCL21.xsd	142
10.2. NCL-AttributeInterface.xsd	144
10.3. NCL-BasicComposite.xsd	144
10.4. NCL-BasicDescriptor.xsd	145
10.5. NCL-BasicLayout.xsd	145
10.6. NCL-BasicMedia.xsd	146
10.7. NCL-BasicTiming.xsd	147
10.8. NCL-component.xsd	147
10.9. NCL-CompositeConnector.xsd	148
10.10. NCL-CompositeDescriptor.xsd	149
10.11. NCL-CompositeInterface.xsd	149
10.12. NCL-compositeTemplate.xsd	150
10.13. NCL-connector.xsd	150

10.14. NCL-ContentControl.xsd	151
10.15. NCL-control.xsd	152
10.16. NCL-CostFunction.xsd	153
10.17. NCL-DescriptorControl.xsd	154
10.18. NCL-interface.xsd	154
10.19. NCL-Language.xsd	157
10.20. NCL-layout.xsd	172
10.21. NCL-link.xsd	174
10.22. NCL-Linking.xsd	175
10.23. NCL-MediaInterface.xsd	176
10.24. NCL-presentation.xsd	177
10.25. NCL-struct.xsd	178
10.26. NCL-Structure.xsd	179
10.27. NCL-SwitchInterface.xsd	179
10.28. NCL-TestRules.xsd	180
10.29. NCL-timing.xsd	180
10.30. NCL-XTemplateUse.xsd	182
11 Apêndice C	183
11.1. XConnector21.xsd	183
12 Apêndice D	195
12.1. XT-BasicConstraints.xsd	195
12.2. XT-BasicLinking.xsd	195
12.3. XT-BasicResources.xsd	196
12.4. XT-BasicVocabulary.xsd	197
12.5. XT-connector.xsd	197
12.6. XT-ConnectorVocabulary.xsd	198
12.7. XT-constraints.xsd	198
12.8. XTemplate21.xsd	199
12.9. XT-language.xsd	200
12.10. XT-linking.xsd	206
12.11. XT-resources.xsd	207
12.12. XT-struct.xsd	207

12.13. XT-Structure.xsd	208
12.14. XT-vocabulary.xsd	209

Lista de figuras

Figura 1:1. Subsistemas de um sistema hipermídia	16
Figura 2:1 - Aplicação SGML.	25
Figura 2:2 - Exemplo de um documento SMIL.	29
Figura 2:3 - Exemplo de um documento NCL.	30
Figura 2:4 - Exemplo de elos NCL e de reuso de conectores.	31
Figura 2:5 - Exemplo do uso de <i>templates</i> de composição.	33
Figura 3:1 - Exemplo de funções de custo em NCL 2.1.	35
Figura 3:2 - Exemplo de um nó <i>switch</i> em NCL 2.0	37
Figura 3:3 - Exemplo de regras de apresentação e nó <i>switch</i> em NCL 2.1.39	
Figura 3:4 - Exemplo de documento NCL 2.1.	42
Figura 3:5 - <i>Switch</i> de conteúdo e de descritores.	44
Figura 3:6 - Máquina de estados de um evento.	47
Figura 3:7 - Exemplos de conectores em NCL 2.0.	49
Figura 3:8 - Relações <i>finishes</i> e <i>overlaps</i> .	49
Figura 3:9 - Exemplo de conectores em NCL 2.1.	50
Figura 3:10 - Exemplo de <i>template</i> de composição em NCL 2.0.	53
Figura 3:11 - Visão temporal de uma composição NCL.	53
Figura 3:12 - Exemplo de <i>template</i> de composição em NCL 2.1.	59
Figura 3:13 - Visão temporal de uma composição NCL.	60
Figura 3:14 - Exemplo de <i>template</i> com relações de inclusão e por conectores em NCL 2.1.	61
Figura 4:1 - Visão temporal de uma composição <i>par</i> em SMIL.	66
Figura 4:2 - Visão estrutural de uma composição XT-SMIL <i>par</i> antes e após o processamento de <i>template</i> .	66
Figura 4:3 - <i>Template audioComLegendasEnPt</i> em XT-SMIL.	68
Figura 4:4 - Composição XT-SMIL <i>par</i> utilizando um <i>template</i> .	70
Figura 4:5 - Resultado do processamento de <i>template</i> em uma composição XT-SMIL <i>par</i> .	70
Figura 4:6 - Exemplo de elo em uma composição SMIL.	71

Figura 4:7 - Exemplo de elo em uma composição XC-SMIL.	72
Figura 4:8 - Exemplo de elos multiponto em uma composição XC-SMIL.	72
Figura 4:9 - Exemplo de uma composição X-SMIL.	73
Figura 5:1 - . Visão geral da estruturação em dois níveis dos <i>frameworks</i> para compiladores de linguagens modulares.	78
Figura 5:2 - Exemplo de um método do tipo <i>parse</i> de um <i>framework</i> de compiladores.	81
Figura 5:3 - Diagrama de classes do gerador automático de <i>frameworks</i> de compiladores.	83
Figura 5:4 - Diagrama de classes do <i>framework</i> para compiladores NCL.	86
Figura 5:5 - Exemplo simplificado de documento NCL.	87
Figura 5:6 - Compiladores NCL.	90
Figura 5:7 - Composição SMIL gerada a partir do compilador NCL-SMIL.	92
Figura 5:8 - Diagrama de classes do Framework para Compiladores SMIL.	9
Figura 5:9 - Processador de <i>Template</i> XTemplate 2.1.	97
Figura 5:10 - Documento XML com a cópia de uma composição sendo processada.	98
Figura 5:11 - Composição gerada pelo processador de <i>templates</i> .	99
Figura 5:12 - Exemplo de transformada <i>body XSLT</i> .	101
Figura 5:13 - Exemplo de transformada <i>link XSLT</i> .	104
Figura 5:14 - Exemplo de transformada <i>constraint XSLT</i> .	105
Figura 6:1 - Exemplo de um documento representando um noticiário no sistema LAMP.	108
Figura 6:2 - <i>Template</i> para artigos de um noticiário.	109
Figura 6:3 - API DOM de JAXP.	110
Figura 6:4 - API SAX de JAXP.	111
Figura 6:5 - Ferramentas XANTLR e TDOM.	114
Figura 6:6 - Especificação da classe <i>FirstApplet</i> em Java.	115
Figura 6:7 - Especificação de <i>FirstApplet</i> em JavaML.	116
Figura 6:8 - <i>Framework</i> XVM para o desenvolvimento de aplicações com XML.	117
Figura 6:9 - Edição de um documento em <i>timeline</i> .	121

Lista de tabelas

Tabela 1 - Elementos do módulo <i>CostFunctions</i> .	35
Tabela 2 - Elementos do módulo <i>TestRules</i> .	38
Tabela 3 - Diferenças entre NCL 2.1 e NCL 2.0.	46
Tabela 4 - Nomes das transições para a máquina de estados de um evento.	47
Tabela 5 - Elementos da linguagem XTemplate de NCL 2.0.	52
Tabela 6 - Elementos da linguagem XTemplate de NCL 2.1.	56
Tabela 7 - Tabela de Elementos.	77
Tabela 8 - Tabela de Grupos de Elementos.	77
Tabela 9 - Tabela de Áreas Funcionais.	77