

Autoria de Documentos Multimídia Baseada na Identificação e Preenchimento de Estruturas Recorrentes

Bruno Seabra Lima¹

Carlos de Salles Soares Neto^{1,2}

Roberto Gerson de Albuquerque Azevedo^{1,2}

¹Laboratório Telemídia - Departamento de Informática – PUC-Rio
Rua Marquês de São Vicente, 225
Rio de Janeiro – 22453-900 – Brasil

²Laboratory of Advanced Web Systems – Departamento de Informática - UFMA
Av. dos Portugueses s/n - Campus do Bacanga
São Luís/MA – CEP: 65085-580 - Brasil

{bslima, csalles, robertogerson}@telemidia.puc-rio.br

RESUMO

A crescente demanda por aplicações multimídia, como as de cunho interativo para TV Digital, requer mecanismos para reduzir o tempo de desenvolvimento de novas aplicações. Uma das formas de reduzir esse tempo é a utilização da autoria orientada a arquétipos como base para a construção rápida de aplicações. Tais arquétipos são documentos incompletos compostos por lacunas, que são partes editáveis passíveis de modificação pelo autor de uma nova aplicação. Este artigo propõe um esquema declarativo de anotações em documentos multimídia que identifique estruturas recorrentes e facilite o preenchimento das lacunas. Um processador de arquétipos foi implementado de forma a permitir a geração de uma instância de um arquétipo baseado em dados fornecidos pelo autor.

Categories and Subject Descriptors

1.7.2 [Document Preparation]: Format and notation, hypertext/hypermedia, Markup Languages, Standards.

General Terms

Documentation, Design, Languages

Keywords

Documentos Multimídia, NCL, Arquétipos

1. INTRODUÇÃO

Nos últimos anos, as aplicações multimídia têm sofrido constante evolução, tanto nos recursos oferecidos, quanto no processo de autoria. Isso levou a utilização desse tipo de aplicação em diversos nichos tecnológicos, dentre os quais Web, IPTV e TV Digital. Essa expansão acarretou o aumento de profissionais trabalhando no desenvolvimento de tais aplicações, assim como o investimento em mecanismos que auxiliem esse processo de

autoria.

Essas aplicações são compostas por várias mídias (áudio, vídeo, texto, imagem, etc.) e seus relacionamentos de sincronismo espaço-temporal. Em um ambiente de TV Digital, seja ele terrestre, a cabo ou sobre uma rede IP, há sempre um fluxo audiovisual contínuo sendo transmitido. As aplicações multimídia são construídas para estarem em sincronismo com esse fluxo. Em programas de TV, onde as aplicações devem ser editadas, ou até mesmo criadas em tempo real, é necessário competitivamente que o processo de autoria seja rápido e eficiente.

Uma das formas para acelerar esse processo de autoria é investir em ferramentas de autoria poderosas [1][2]. Tais ferramentas auxiliam o autor dessas aplicações em vários aspectos, muitas vezes automatizando alguns passos no processo de autoria. Ao trabalhar com essas aplicações, nesses diferentes nichos, é fácil perceber que seus códigos-fonte compartilham as mesmas estruturas recorrentes. Sendo assim, outra maneira de acelerar a autoria é identificar quais são as partes comuns a essas aplicações e proporcionar ao autor modificar somente elementos presentes nessas estruturas recorrentes. Esse processo é chamado de autoria orientado a arquétipos [3].

Uma premissa deste trabalho é a existência de várias famílias de documentos multimídia com estruturas recorrentes em comum. Essas estruturas podem ser de composição, apresentação ou dos relacionamentos entre as mídias que compõem a aplicação. Tais características possibilitam a criação de uma ampla gama de arquétipos de aplicações. Por exemplo, vários arquétipos de EPG (Electronic Program Guide) podem ser criados, dentre os quais: um EPG que preserva as mesmas mídias e muda somente o leiaute, facilitando a atualização rápida de design; ou um EPG que preserva o leiaute e muda as mídias, auxiliando o desenvolvimento rápido de um EPG para um novo canal.

Para que uma aplicação multimídia possa servir como arquétipo para a autoria de outras aplicações é necessário criar um mecanismo para anotar quais são as partes editáveis, também chamadas de lacunas. Além disso, o autor que está utilizando o arquétipo deve ser capaz de expressar os dados para o preenchimento das lacunas, os quais o tornam único. Este artigo propõe uma notação declarativa para identificação dessas lacunas

e um processador de arquétipos capaz de gerar rapidamente novas aplicações multimídia.

A Seção 2 mostra os diversos trabalhos relacionados existentes na literatura e discute as vantagens e desvantagens do seu uso no contexto de aplicações multimídia. A Seção 3 apresenta uma proposta para o uso de arquétipos na autoria dessas aplicações. A Seção 4 mostra um estudo de caso da aplicação da proposta na construção de uma aplicação multimídia dada como exemplo. Finalmente, a Seção 5 discute as vantagens e desvantagens da proposta e os trabalhos futuros.

2. TRABALHOS RELACIONADOS

Fazendo uma analogia à definição de arquétipo dada por Singh [4], no contexto de aplicações multimídia, um arquétipo representa um conjunto de características e atributos que podem especificar, parcialmente ou completamente, a composição estrutural, espacial e semântica de uma aplicação. Dentro dessa definição são encontradas na literatura propostas para o uso de arquétipos na autoria de aplicações multimídia.

A geração dinâmica de páginas XHTML [5] é feita mediante o preenchimento de lacunas nas páginas com conteúdos recuperados a partir de bancos de dados. Para dar suporte a essas tecnologias, como os Servlets, as páginas são geradas imprimindo blocos de código XHTML intercalados com comandos típicos da linguagem. Um problema claro dessa abordagem é que os designers gráficos não conseguem fazer seu trabalho sem entender essas linguagens, acarretando um grau maior de erros na programação ou no próprio leiaute final.

Uma solução é fazer uma inversão, embutindo o código imperativo dessas linguagens no XHTML. Dessa maneira, o designer gráfico pode se preocupar somente com o leiaute das páginas XHTML ignorando os dados que serão apresentados. Nessa abordagem, ainda existe o inconveniente de misturar a lógica do programa intimamente com o leiaute. Isso diminui a flexibilidade, dificulta o processo de desenvolvimento e aumenta os custos de manutenção.

De acordo com Parr [6], existem diversos motivos para forçar essa separação. Alguns desses motivos podem ser destacados: encapsulamento (o leiaute e lógica são entidades completas e independentes), clareza (o arquétipo é um XHTML legível por um designer ou programador), reúso (leiautes podem ser reaproveitados). Devido a esses benefícios, foi encorajada a utilização de arquétipos compostos de partes comuns, no caso código XHTML, e partes variáveis, como exemplo código PHP. Há produtos [7], chamados de *template engines*, que provêem um mecanismo para a utilização de arquétipos no desenvolvimento de páginas XHTML dinâmicas.

Linguagens multimídia, como SMIL[8] e NCL[9], têm o foco no sincronismo espaço-temporal, diferente de XHTML, que tem o foco na apresentação espacial. NCL e SMIL acrescentam a dimensão de tempo, incorporando declarativamente uma semântica de comportamento inerente nesse tipo de aplicação e, por isso, são classificadas como linguagens declarativas baseadas no tempo.

As propostas para utilização de arquétipos vistas até aqui não funcionam em tais aplicações, porque, diferente do XHTML, parte da lógica dessas aplicações devem ser especificadas pela própria linguagem multimídia. Um exemplo claro disso são os contêineres temporais de SMIL: o *seq* tem um comportamento de

seqüência temporal entre os seus objetos de mídias; o *par* dá a idéia de paralelismo entre suas mídias. NCL vai além de SMIL, utilizando a noção de relacionamentos espaço-temporais genéricos, onde cada mídia pode assumir um papel nesse relacionamento. Por exemplo, “depois de 10s de exibição da imagem X é iniciada a exibição do vídeo Y”.

Apresentadas essas dificuldades, há propostas na literatura [10][11] que fornecem mecanismos para o uso de arquétipos na autoria dessas aplicações. Tais propostas visam uma linguagem para definição de arquétipos, onde o comportamento temporal de partes da aplicação é abstraído dos objetos de mídia declarados na linguagem multimídia alvo. Essa não é uma abordagem ideal, pois, de acordo com Parr [6], se a linguagem do arquétipo é muito poderosa, os designers de arquétipos ficam tentados a adicionar toda a lógica de negócios onde precisam.

É importante não separar a lógica temporal da especificação dos objetos de mídia. Isso porque um autor, ao utilizar um arquétipo para a construção de uma nova aplicação, deve ser capaz de entender a lógica espaço-temporal ali contida. Uma forma direta de atender a esses requisitos é embutir no próprio documento multimídia um mecanismo para a anotação declarativa das partes que podem ser modificadas. Isso é o que diferencia essa proposta das demais encontradas na literatura. Na próxima seção é apresentada uma proposta que visa atender a esses requisitos.

3. PROPOSTA

As aplicações multimídia com muitos relacionamentos espaço-temporais tendem a ser bastante prolixas. Especialmente, em NCL essa prolixidade se dá pela declaração repetitiva de elos que referenciam o mesmo conector, mas com objetos de mídia diferentes. Tal repetição de código é mais facilmente identificada em aplicações baseadas em certos conjuntos de mídias que interpretam um papel semântico parecido.

Um exemplo é uma aplicação de *slideshow* de fotos com legenda. As mídias que compõem essa aplicação se relacionam espaço-temporalmente de forma parecida, diferindo apenas em algumas peculiaridades. Ao isolar essas peculiaridades das partes comuns, o autor de um novo *slideshow* pode se preocupar apenas em fornecer quais são as fotos e legendas.

Nesta proposta um arquétipo é um documento multimídia anotado. Essa anotação é feita utilizando uma notação declarativa baseada em conjuntos e delimitada por chaves, $\{ \}$, passando a idéia de lacunas. Diferente das propostas apresentadas na Seção 2, esta abordagem não separa a semântica temporal em outro documento, apenas cria um mecanismo para evitar a declaração redundante dessa semântica temporal em documentos multimídia. Dessa forma, o usuário que está utilizando o arquétipo não terá uma grande carga cognitiva para entender a lógica contida naquele arquétipo.

A Figura 1 apresenta os atores e documentos envolvidos na proposta. Como pode ser visto na figura, tem-se o papel do “Autor de Arquétipos” que é responsável por identificar e anotar as partes do documento multimídia passíveis de modificação. O “Autor de Aplicações” utiliza o documento multimídia anotado como arquétipo base para criação de uma nova aplicação.

O “Autor de Aplicações” só precisa fornecer os dados, partes editáveis, que irão completar o arquétipo. Este artigo ainda propõe um processador de arquétipos que tem como entrada o documento multimídia anotado e os dados fornecidos pelo “Autor de

Aplicações” gerando como resultado a nova aplicação multimídia pronta para ser apresentada, conforme detalhado na Figura 1.

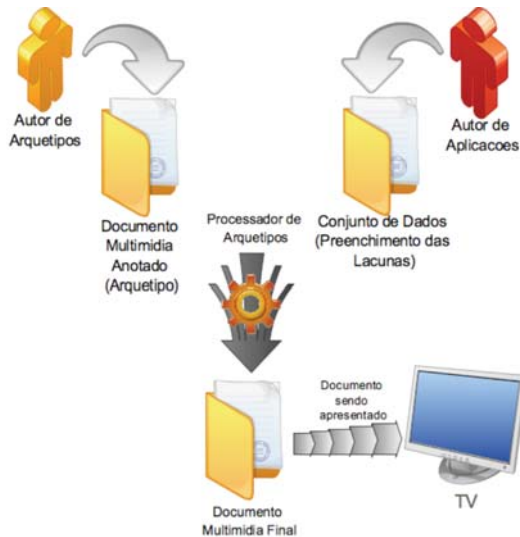


Figure 1 – Atores e documentos no processo de autoria proposto.

3.1 Notação Declarativa

O documento multimídia anotado é um documento escrito em NCL (ou SMIL), no qual em alguns atributos existem lacunas delimitadas por chaves ({}). A notação utilizada nessas lacunas indica como esses atributos serão preenchidos no documento multimídia final. Com o objetivo de diminuir a carga cognitiva da proposta, a notação utiliza o paradigma declarativo, assim como a maioria das linguagens multimídia. O mesmo paradigma declarativo foi utilizado no documento em que o “Autor de Aplicações” especifica os dados para o preenchimento das lacunas.

Essa notação é baseada em lógica de conjuntos. Para acessar uma informação que está contida no documento de dados do “Autor de Aplicações”, o “Autor de Arquetipos” utiliza a seguinte notação: **ELEMENTO**[**EXPRESSÃO**].**ATRIBUTO**. O **ELEMENTO** identifica o nome de um elemento XML no conjunto de dados. Logo após, tem-se uma expressão matemática ([**EXPRESSÃO**]) com constantes numéricas ou nomes de blocos para identificar um elemento específico dentre os vários declarados no conjunto. O **ATRIBUTO** referencia o nome do atributo do elemento que está sendo indexado. As informações obtidas por essa tripla de valores substituirão a lacuna demarcada no documento multimídia final. Adicionalmente, o valor de **ATRIBUTO** pode ser uma palavra reservada (**auto**), sinalizando ao processador de arquetipos que tal lacuna deve ser preenchida com um identificador único no documento final.

Resolvido o problema da identificação dos elementos no conjunto, se faz necessário marcar as estruturas recorrentes no documento. Isso é feito com o uso do elemento *block*, que define uma variável (atributo *name*) utilizada na indexação dos elementos do conjunto. Os elementos filhos deste elementos serão tratadas pelo processador como um bloco de iteração. Em cada iteração, os atributos demarcados ({}), naquele bloco terão sua expressão resolvida e serão adicionados ao documento final.

Na marcação do bloco pode ser preciso limitar o número de iterações que serão realizadas, permitindo ao “Autor de Arquetipos” definir famílias de documentos que possuam restrições de cardinalidade. Para isso, são utilizados os atributos *initialValue* e *finalValue*. Esses atributos são definidos como expressões e restringem o número mínimo e máximo de interações sobre aquele bloco.

A Seção 4 apresenta um estudo de caso que esclarece os mecanismos utilizados nessa proposta. Um *slideshow* de fotos é apresentado como um documento multimídia anotado, neste caso escrito em NCL, junto com um exemplo de documento com os dados fornecidos pelo “Autor de Aplicações”.

4. ESTUDO DE CASO

Como estudo de caso é utilizada uma aplicação multimídia de *slideshow* de fotos escrita em NCL. Essa aplicação é dada por uma seqüência de imagens que são exibidas em intervalos de tempo iguais, sempre com a próxima imagem começando a ser exibida assim que a anterior termina sua exibição. A Figura 2 ilustra esquematicamente um *slideshow* simples com um título sendo exibido durante a apresentação das fotos.

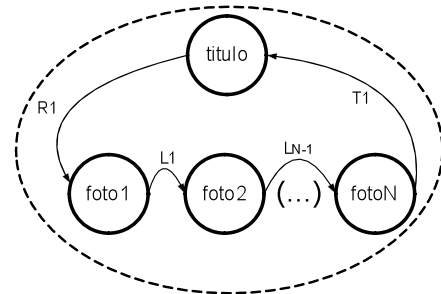


Figura 2 – Slideshow de fotos com um título.

Segundo o conhecimento dos autores, não há linguagens de autoria multimídia de propósito geral que possam modelar diretamente um *slideshow* de fotos. Isso porque, habitualmente não é possível em tais linguagens expressar um documento como um conjunto de tamanho indefinido de objetos de mídia (nesse caso, o número de fotos é variável).

A Listagem 1 mostra um exemplo de definição do documento NCL anotado que representa o *slideshow*, segundo a notação proposta neste trabalho. É fácil perceber que se o autor criasse diretamente um documento NCL, para cada par de fotos, em seqüência, a estrutura apresentada nas linhas de 9 a 12 deveriam ser replicadas.

Com as estruturas recorrentes devidamente anotadas, fica a cargo do “Autor de Aplicações” especificar um documento (*dataSource*) que contenha apenas os dados necessários para o preenchimento das lacunas, como exemplificado na Listagem 2. A nomenclatura usada no *dataSource* é inferida a partir dos nomes dos **ELEMENTOS** utilizados pelo “Autor de Arquetipos” para a marcação do documento anotado.

```

1: <context id="slideshow">
2: <port id="pPhoto1" component="{photo[1].auto}" />
3: <port id="pTitle" component="title" />
4:
5: <media id="title" src="{title[1].src}" descriptor="dsTitle" />
6: <block name="it">
7: <media id="{photo[it].auto}"
8:   src="{photo[it].src}" descriptor="dsPhoto" />
9: </block>
10: <block name="it">
11: <link xconnector="onEndStart" >
12:   <bind role="onEnd" component="{photo[it].auto}" />
13:   <bind role="start" component="{photo[it+1].auto}" />
14: </link>
15: <link xconnector="onEndStop">
16:   <bind role="onEnd" component="{photo[#photo].auto}" />
17:   <bind role="stop" component="title" />
18: </link>
19: </block>
20: </context>

```

Listagem 1 – Documento NCL anotado para o exemplo do *slideshow*.

```

1: <dataSource>
2:   <title src="title.html" />
3:
4:   <photo src="photo01.png" />
5:   <photo src="photo02.png" />
6:   <photo src="photo03.png" />
7: </dataSource>

```

Listagem 2 – DataSource para o exemplo do *slideshow*.

5. CONCLUSÃO

A proposta foi testada e se mostrou eficiente para uma grande variedade de aplicações multimídias. Dentre as quais, uma aplicação de “Jogo da Velha” desenvolvida totalmente em NCL e disponível no ClubeNCL¹. Inicialmente, o código da aplicação tinha 839 linhas. Após a identificação das estruturas recorrentes e utilização da notação proposta o número de linhas foi reduzido para 305 linhas. A partir do arquétipo construído pode-se facilmente criar um “Jogo da Velha” como uma matriz 4x4, 5x5, etc. Esse exemplo, assim como a implementação do processador de arquétipos encontra-se disponível como código livre².

A notação utilizada para a marcação das lacunas tem uma baixa carga cognitiva e não separa o arquétipo do documento multimídia, diferente das propostas vistas na Seção 2. O arquétipo continua sendo um documento multimídia, apenas está incompleto. Como principal vantagem, essa abordagem possibilita que as estruturas recorrentes só necessitem ser especificadas uma vez, ficando a cargo do processador de arquétipos gerar o documento final.

As ferramentas de autoria podem ser adaptadas para tratar esse arquétipo como um documento multimídia normal. Assim como o “Autor de Aplicações” é capaz de ler o documento anotado e entender a lógica do programa, essas ferramentas podem interpretar a notação e fornecer ao seu usuário uma abstração

desse documento, substituindo as partes editáveis por mídias padrões.

Utilizar essa proposta para a implementação de uma ferramenta de autoria baseada em arquétipos sobre a ferramenta *Composer* [1], utilizada para criação de documentos NCL, é visto como um trabalho a ser realizado. Pretende-se, também, aumentar a expressividade da notação de forma a torná-la mais abrangente.

6. REFERÊNCIAS

- [1] Guimarães, R. L., Costa, R. M. R., and Soares, L. F. G. 2007. *Composer: Ambiente de Autoria de Aplicações Declarativas para TV Digital*. XIII Simpósio Brasileiro de Sistemas Multimídia e Web – WebMedia2007. Gramado, Brasil – Outubro de 2007.
- [2] Deltour, R. and Roisin, C. 2006. The limsee3 multimedia authoring model. In *Proceedings of the 2006 ACM Symposium on Document Engineering* (Amsterdam, The Netherlands, October 10 - 13, 2006). DocEng '06. ACM, New York, NY, 173-175. DOI= <http://doi.acm.org/10.1145/1166160.1166203>
- [3] C.S.S. Neto and L.F.G. Soares. 2008. Autoria orientada a arquétipos para TV digital: uma abordagem restritiva e direcionada. XXXIV Conferencia LatinoAmericana de Informatica. (Santa Fé, Argentina)
- [4] Singh, A. and Schaeffer, J. and Szafron, D. 1996. Views on Template-Based Parallel Programming. In *Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research* (Toronto, Ontario, Canada). p. 35.
- [5] W3C. 2000. XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition) – W3C Recommendation. Janeiro 2000. Disponível em <http://www.w3.org/TR/html/>.
- [6] Parr, T. J. 2004. Enforcing strict model-view separation in template engines. In *Proceedings of the 13th international Conference on World Wide Web* (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM, New York, NY, 224-233. DOI= <http://doi.acm.org/10.1145/988672.988703>
- [7] HTML Template Sigma. Disponível em: http://pear.php.net/package/html_template_sigma
- [8] W3C. 2008. Synchronized Multimedia Integration Language (SMIL) 3.0 — W3C Recommendation. Dezembro 2008. Disponível em <http://www.w3.org/TR/2008/REC-SMIL3-20081201/>
- [9] ABNT - Associação Brasileira de Normas Técnicas (2007) — Televisão Digital Terrestre- Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 2: Gíngua-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações.
- [10] Celentano, A. and Gaggi, O. 2003. Template-based generation of multimedia presentations. *Journal of Software Engineering and Knowledge Engineering*, v. 13, n. 4, p. 419-445.
- [11] Geurts, J. P., Ossenbruggen, J. R., and Hardman, L. 2001 *Application-Specific Constraints for Multimedia Presentation Generation*. Technical Report. UMI Order Number: INS-R0107., CWI (Centre for Mathematics and Computer Science)

¹ <http://www.clube.ncl.org.br>

² <http://www.telemidia.puc-rio.br/~bslima/site/ncltemplates.html>