

Autoria orientada a arquétipos para TV digital: uma abordagem restritiva e direcionada

Carlos de Salles Soares Neto^{1,2} e Luiz Fernando Gomes Soares¹

¹ Departamento de Informática, PUC-Rio, Caixa Postal 38.907 - CEP 22.453-900 – Rio de Janeiro, RJ – Brasil

² Departamento de Informática, UFMA. Av. dos Portugueses, S/N – Campus do Bacanga CEP 65085-580 – São Luís, MA – Brasil
csalles@deinf.ufma.br, lfgs@inf.puc-rio.br

Abstract. Many multimedia documents, specially those representing interactive Digital TV applications, have a similar presentation and composition structure, although they follow different archetypes. This paper explores these similarities, proposing a new approach called archetype oriented programming that drives the author in the creation of new documents based on the specialization or instantiation of existing archetypes. These archetypes represent document classes with related vocabulary, semantics or contents and with some points that are often changed among them. By the proposed approach, the user is driven to take a constrained set of decisions in order to create new documents in a faster and simpler way when compared to general purpose techniques. As a concept proof, the NCLWizard tool is presented to allow the semi-automatic authoring of NCL documents.

Keywords: Digital TV, Nested Context Language, NCLWizard, Archetype Oriented Programming.

1 Introdução

A implantação de um sistema de TV digital interativo cria impactos tecnológicos bem mais amplos do que apenas melhorar sensivelmente a qualidade da imagem e do som. Há uma tendência natural ao surgimento de uma vasta gama de novos serviços, como o oferecimento de guias eletrônicos de programas, a distribuição de jogos de computador, o controle de acesso, a proteção de conteúdo (DRM), vídeo sob demanda (VoD), aplicações de comércio eletrônico e jornais eletrônicos. Notadamente, esse conjunto de funcionalidades pode ser resumido como agregar capacidade computacional à TV.

Adicionalmente, a produção de conteúdo para TV digital também sofre uma mudança brusca em relação à TV analógica. O conteúdo não é mais necessariamente linear. Através da interatividade do usuário, pode haver múltiplas formas de assistir a um mesmo conteúdo. A produção de conteúdo não-linear exige mudanças em quase todas as fases de construção do mesmo.

Em um roteiro não-linear, em cada ponto de interatividade, o roteirista precisa especificar as quebras na forma seqüencial de especificação de um programa, definindo novas linhas de tempo distintas, dependendo da decisão do telespectador, o que exige um formato com redirecionamento. Quando utilizado, o *storyboard* também precisa refletir esse novo formato. Na prática, o programa não é definido por uma linha do tempo única e sim por uma cadeia principal de eventos e por algumas cadeias secundárias que dependem da interação do usuário.

Em meio a essa aparente quebra de paradigma na produção de conteúdo para a TV digital interativa é necessário enfatizar que o processo de autoria ainda precisa ser rápido, eficiente e simples.

A construção rápida de programas interativos é uma característica e demanda estratégica. O estreitamento do tempo necessário para a elaboração de novos serviços é vital em um cenário competitivo e dinâmico como o de produção de conteúdo. Uma possível vertente para atender a esse requisito é o aparecimento de ferramentas de autoria especializadas na construção de programas em formatos específicos como telejornais, *talking shows*, *game shows* etc. Idealmente, o processo de autoria do conteúdo interativo precisa ser incorporado ao processo de edição e pós-produção.

A eficiência na autoria também é um requisito importante e tem a ver com a facilidade em criar novos programas interativos de forma correta. Há vários artifícios que mantêm elevada a eficiência, independente do tempo de desenvolvimento ser curto ou longo. Alguns exemplos disso são a utilização de verificadores de código, a existência de um ambiente integrado de desenvolvimento e o uso de linguagens de autoria de alto nível de abstração que conduzem ao resultado final com muito menos linhas de código.

A simplicidade na autoria de programas interativos implica diretamente em uma redução significativa no tempo e custo de aprendizado. Um outro aspecto importante é que quanto mais simples é a autoria, mais abrangente ela se torna, possivelmente enquadrando diversos perfis profissionais diferentes. Ainda é cedo para prever de que forma o ambiente de criação de conteúdo digital se altera com a inserção de programas não-lineares, mas é possível que um novo perfil profissional surja: o autor de programas interativos. Esse profissional terá a tarefa de desenvolver o conteúdo interativo em parceria com a equipe de edição e com base em alguns artefatos recebidos dos roteiristas. Idealmente, se o processo de autoria for simples o bastante, os próprios roteiristas ou equipe de edição podem assumir esse papel, o que acentua ainda mais a importância do processo ser simples.

O trabalho proposto neste artigo visa direcionar o autor a conseguir o resultado esperado de forma mais rápida, simples e eficiente. A abordagem proposta restringe o autor a criar um novo documento de acordo com uma classe específica de aplicações de TV digital, que são chamadas de arquétipos de documentos. A complexidade no processo de autoria é separada em dois níveis. De um lado, o autor de arquétipos precisa conhecer a linguagem alvo do documento e identificar estruturas de programas que se repetem. De outro lado, o autor de documentos é restrito a alguns parâmetros que precisam ser preenchidos para instanciar o arquétipo. A abordagem proposta visa também tornar a autoria passível de ser desenvolvida pelos próprios profissionais de TV sem necessidade de aptidões prévias.

O principal objetivo deste trabalho é caracterizar um método para a criação de programas para TV digital interativa, dando-lhe o nome de autoria orientada a

arquétipos. Adicionalmente, é proposta uma nomenclatura própria envolvendo certos atores, bases de documentos e ferramentas que tornam claro o método e sua aplicabilidade. O artigo tem a seguinte organização. A Seção 2 oferece uma visão geral da proposta deste artigo. Como prova de conceito, a Seção 3 descreve a ferramenta NCLWizard, que é um caso de uso baseado na aplicação de *wizards*. A Seção 4 apresenta as conclusões e trabalhos futuros.

2 Autoria Orientada a Arquétipos Em Ambientes de TV Digital

O processo de autoria para TV digital é marcadamente dividido em abordagens ditas imperativas e declarativas. No desenvolvimento imperativo, o programa é descrito de forma algorítmica, como uma seqüência de passos, onde a engenharia de *software* já é bastante madura em termos de pesquisa voltada para aumentar a produtividade e reuso [10]. Alguns exemplos disso são os usos de padrões de projeto [9], frameworks orientados a objetos [6] e programação orientada a aspectos [11]. Geralmente Java é a linguagem utilizada nos paradigmas imperativos para garantir maior interoperabilidade [13].

No âmbito declarativo, é a área de autoria em sistemas hipermídia e multimídia que apresenta soluções para guiar o processo de desenvolvimento com abstrações e estruturas de alto nível [3] [8]. O ambiente declarativo define uma máquina abstrata com um foco específico sobre a qual as aplicações são executadas. No caso de TV digital, esse foco é o sincronismo de mídias, adaptabilidade e suporte a múltiplos dispositivos. Exemplos de linguagens declarativas usadas no contexto de TV digital são Nested Context Language (NCL) [1] e Broadcast Markup Language (BML) [2].

Em vários padrões de TV Digital, os dois tipos de aplicações coexistem, caso em que é conveniente que o middleware existente no dispositivo receptor integre essas duas abordagens. Note que esse enfoque é importante por determinar que aplicações interativas para TV digital nada mais são do que um tipo específico de documento multimídia cujas relações de sincronismo geralmente estão centradas no vídeo e áudio principais.

Ao se analisar um conjunto considerável e aleatório de documentos multimídia oriundos de vários autores, é comum verificar a repetição de uma mesma estrutura de apresentação temporal e espacial, independente daquela aplicação se tratar de uma aplicação imperativa ou declarativa. É difícil identificar os vários motivos pelos quais se observam essas mesmas estruturas básicas em vários documentos diferentes, mas provavelmente há duas razões notórias: primeiro, porque a maioria dos autores baseia-se nos mesmos tutoriais para aprender novas linguagens multimídia e acabam, inevitavelmente, se limitando a explorar as mesmas funcionalidades e estrutura de cada linguagem, segundo, porque há a tendência a copiar o mesmo tipo de aplicação ou programa, só substituindo os conteúdos, ou seja, uma aplicação criativa é rara.

O agrupamento desses documentos segundo suas estruturas básicas em comum possibilita a definição do que pode ser chamado de arquétipo de documento multimídia, também chamado de *template*, estilo etc. Esse agrupamento permite que passos repetitivos no processo de criação dos documentos possam ser automatizados de forma que o autor precise apenas completar algumas lacunas em aberto ao invés de

criar o documento por inteiro. Informalmente, arquétipos podem ser vistos como documentos quase prontos que ainda requerem o preenchimento de alguns pontos em aberto que os tornam únicos.

A existência de arquétipos de documentos é utilizada em vários nichos, como na criação de páginas da internet ou em editores de texto. Os arquétipos de documentos também são explorados por quase a totalidade das linguagens de autoria multimídia existentes. NCL [1], por exemplo, faz uso da composição de nós e elos para aumentar o reuso no processo de autoria. Uma forma rudimentar de expressar esses arquétipos é exatamente através do uso de nós de composição. Há, entretanto, outras formas mais sofisticadas exigindo, contudo, um conhecimento mais aprofundado sobre a linguagem de autoria, tanto para quem cria o arquétipo quanto para quem faz uso dele. Os *templates* NCL [15] são um exemplo disso.

Estima-se que grande parte das páginas *web* atualmente é desenvolvida por não-programadores. Uma prática que era muito comum há alguns anos e que ainda resiste até hoje é a de montar o *design* da própria página a partir de fragmentos extraídos de outras páginas. Com o advento de aplicações *web* dinâmicas, um novo modelo de concepção de páginas foi pouco a pouco sendo criado. Meta-aplicações, que servem para a configuração de um tipo específico de aplicações, começaram a surgir. Tais aplicações permitiram que uma grande massa de usuários finais pudesse submeter conteúdo próprio na internet sem necessidade de conhecimentos prévios de programação de qualquer tipo. Um exemplo disso é o advento explosivo da quantidade e diversidade dos *blogs* e *fotologs* nos últimos anos. Provavelmente a principal lição retirada desses casos de sucesso na internet é que o processo de autoria precisa ser aproximado do usuário final de tal forma que a concepção de conteúdo com aparência profissional precisa ser simplificada e rápida ao ponto que programadores profissionais só sejam indispensáveis para a elaboração da própria ferramenta geradora de aplicações.

É conveniente reafirmar que este trabalho focaliza especificamente na concepção de documentos para TV digital. Programas interativos de TV conduzem a certas simplificações ou especificidades importantes. Em primeiro lugar, a entrada de dados é geralmente resumida ao controle remoto, que não possui mecanismos apontadores ou a facilidade de um teclado alfanumérico. A distância do usuário para a tela da TV é consideravelmente maior que do usuário de computador de seu monitor, o que é uma das principais razões que acarretam na TV ter uma linguagem caracteristicamente composta por muito mais imagens que texto. Essa diferença é importante, pois caracteriza que o desafio é muito menos o de organizar a formatação espacial de dados textuais e sim o de estruturar um conteúdo multimídia com relações de sincronismo espaço-temporais entre si.

Por outro lado, a TV ainda tem uma penetração bem maior que o computador, o que pode conduzir ao raciocínio que os desafios envolvidos no projeto de boas interfaces nesse ambiente são bem maiores porque devem ser mais gerais e não voltadas a nichos específicos. O uso recorrente de um mesmo padrão de interface é uma forma coerente de conduzir o usuário ao fácil reconhecimento do funcionamento de um novo programa [14]. Essa tendência em haver vários documentos com a mesma estrutura também é reforçada pelo fato da TV produzir constantemente os mesmos programas com o mesmo formato, como é o caso de telejornais. Isso enfatiza a aplicabilidade da proposta deste trabalho.

A Figura 1 ilustra diagramaticamente a abordagem de autoria orientada a arquétipos. Com o fim de sistematizar e facilitar a aplicação da presente proposta, dois atores são identificados. O autor de arquétipos inicia o processo construindo um arquétipo de documentos ou uma base deles. Uma vez conhecendo essa base de arquétipos por meio de uma ferramenta ou de abstrações de alto nível, o autor de documentos instancia um arquétipo específico por meio da parametrização de suas lacunas. Seu produto final é um documento multimídia que representa um programa interativo.

Há duas formas do autor de documentos exercer seu papel. A primeira alternativa, ilustrada pela Figura 1(a) é que ele seja um mero usuário da base de arquétipos existentes, sem possibilidade de alterá-la. A segunda alternativa, descrita na Figura 1(b) é que ele seja um elemento atuante sobre a ferramenta de autoria, possivelmente criando novos arquétipos a partir da combinação ou agregação de arquétipos existentes entre si ou ainda por meio da especialização de arquétipos existentes. Na prática, isso os permite conceber novos arquétipos, mas, ainda assim, eles o fazem sempre a partir de uma base previamente definida e não com toda a liberdade do autor de arquétipos.

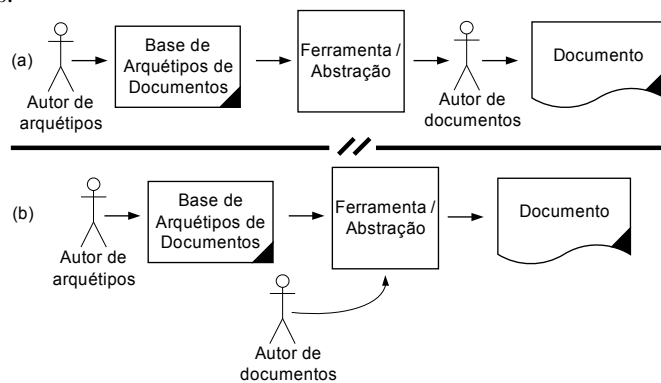


Fig. 1: Autoria orientada a arquétipos. (a) Autor de documentos usuário da base. (b) Autor de documentos atuando sobre a base.

A necessidade de haver, pelo menos conceitualmente, dois papéis separados de autoria se deve ao fato que a motivação principal de uso de arquétipos é facilitar a autoria para quem os aplica e não para quem os cria. De forma mais simples, pode-se dizer que o autor de documentos é o alvo das facilidades propostas enquanto ao autor de arquétipos resta fazer uso de técnicas convencionais. Isso conduz a um fato importante. É sabido que criar o vocabulário e regras de um arquétipo para instanciá-lo em um único documento traria mais do que o dobro de trabalho que fazer o documento diretamente. Ganha-se tempo e facilidade apenas quando um mesmo arquétipo é instanciado em vários documentos diferentes (essa quantidade é difícil de mensurar porque depende de como o arquétipo é especificado e da estrutura que ele representa). Isso não representa um problema, contudo, já que o cenário esperado para o ambiente de autoria de TV digital interativa, conforme previamente traçado, tende a ser repetitivo pelo menos estruturalmente.

Um desafio para a autoria orientada a arquétipos é permitir a comunicação entre os dois tipos de autores. Como o autor de documentos deve ter seu trabalho simplificado, ele é isolado em um nível de abstração mais alto com relação ao autor de arquétipos, que trabalha diretamente na linguagem alvo do documento. A solução para esse problema é o uso de abstrações ricas e possivelmente visuais que permitam a comunicação entre os dois autores. A seção seguinte ilustra um caso de uso em que *wizards* são utilizados com esse fim. Outras abordagens são discutidas como trabalho futuro na Seção 4. É importante reforçar que está fora do escopo deste trabalho especificar formatos de armazenamento e representação únicos para os arquétipos de documento, o que torna bem mais flexível a aplicabilidade da presente proposta. A seção seguinte, por exemplo, representa os arquétipos por meio de documentos NCL ou *templates*, o que poderia não ser satisfatório caso a linguagem alvo seja Java.

3 NCLWizard: um Caso de Uso Baseado em Ferramentas *Wizard*

Um *wizard* é um programa de computador interativo que atua como uma interface para guiar o usuário em uma tarefa complexa, por meio de formulários passo-a-passo. *Wizards* são artefatos bem conhecidos, que podem ser encontrados em muitas aplicações, tais como programas de instalação, caixas eletrônicas, e ainda na automatização do processo de geração de código, sempre que é detectado um processo repetitivo em uma implementação [14]. Em especial, ferramentas de autoria multimídia também podem possuir *wizards* embutidos, para permitir que o usuário especifique um novo documento sem precisar de conhecimento prévio sobre o formato ou linguagem em que o documento final será gerado. Como vantagem adicional, o processo de criação é muito mais rápido, porque é feito de forma semi-automática. No entanto, a expressividade do autor é limitada já que ele se restringe a apenas preencher algumas informações requisitadas pelo *wizard*.

Fazendo uso da nomenclatura proposta neste trabalho, artefatos *wizards* são empregados como ferramenta para a autoria orientada a arquétipos. Os *wizards* atuam como uma interface fácil e amigável entre o usuário e o documento produzido. Cada *wizard* é uma unidade independente, simples e concisa de geração automatizada de código para um arquétipo específico. O usuário de um *wizard*, por sua vez, faz o papel de autor do documento, focalizando apenas no preenchimento das lacunas que tornam aquele documento único em relação aos demais que seguem o mesmo arquétipo. O formato original do arquétipo e a linguagem alvo do documento são completamente abstraídos pelo direcionamento dado pela ferramenta.

A ferramenta NCLWizard foi desenvolvida tendo em vista não apenas a possibilidade de confecção de *wizards* visualmente ricos e abrangentes, com amplas facilidades para o usuário, mas também com a meta de ser fácil a construção de novos *wizards* tão logo sejam definidos novos arquétipos. O NCLWizard é composto por três elementos principais (vide Figura 2): o Gerador e o Interpretador de Interfaces e o Gerador de Documentos. O primeiro elemento é responsável pela leitura do plano de execução e construção da interface com o autor de documentos. O segundo elemento controla a interface propriamente dita com o usuário e repassa as respostas dadas pelo usuário para o gerador de documentos. Finalmente, o terceiro elemento engloba a

geração de código e lida com detalhes específicos da linguagem alvo, instanciando o arquétipo de acordo com as opções e respostas do usuário.

A criação de um novo *wizard* está sempre associada a um novo arquétipo. Esse arquétipo é utilizado como base pela ferramenta para sofrer transformações e ser gerado um documento na linguagem alvo. NCLWizard foi implementado tendo em vista NCL como linguagem-alvo, porém sua arquitetura é genérica o bastante para poder ser aplicada, com alterações exclusivamente no Gerador de Documentos, em outras linguagens multimídia, como SMIL[5]. O único requisito básico é que essa linguagem possua algum mecanismo de identificação unívoca de seus elementos.

O processo de construção de um novo *wizard* pode ser dividido em duas fases distintas: primeiro, é necessário identificar e desenvolver um novo arquétipo; depois, cria-se a interface do *wizard* e são especificadas as transformações que devem ser aplicadas ao arquétipo de acordo com as respostas do usuário, para se gerar o novo documento. A primeira fase é concebida pelo Autor de Arquétipos, conforme descrito na seção anterior. A novidade é que na segunda fase entra em cena o Criador de *Wizards* (geralmente o próprio autor de arquétipos), que utiliza como base o arquétipo proposto para restringir aquilo que o autor pode modificar. Isso é feito por meio de uma Especificação XWizard, como é visto adiante, que descreve tanto a interface passo-a-passo com o autor quanto as respectivas transformações. O processo de criação de *wizards* e o uso da ferramenta pelo Autor de Documentos é diagramaticamente apresentado na Figura 2.

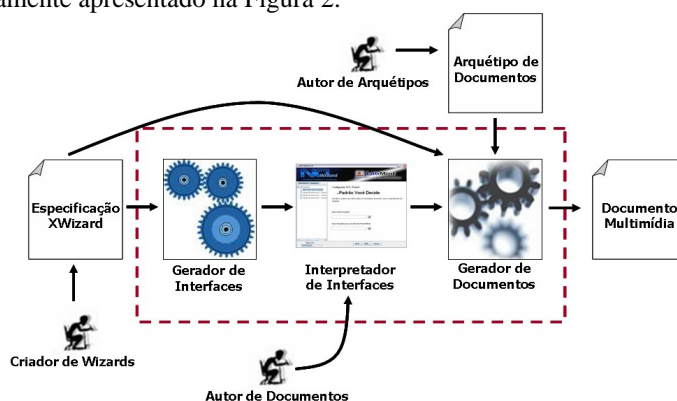


Fig. 2: Atores e Arquitetura da ferramenta NCLWizard.

Com o objetivo de especificar novos *wizards*, uma aplicação XML chamada XWizard foi desenvolvida. A vantagem de se escolher essa abordagem é que assim a ferramenta NCLWizard permite que novos *wizards* sejam acoplados muito rapidamente sem precisar alterar o código da ferramenta. Uma especificação XWizard descreve como se dá a interface com o usuário e quais elementos e atributos XML do arquétipo de documentos relacionado podem ser modificados de acordo com as respostas dadas pelo autor. Esse plano de execução descrito em XWizard é interpretado para compor dinamicamente a interface com o autor bem como gerar um novo documento com base nas respostas do autor.

A Tabela 1 lista os elementos e atributos de XWizard. Em função da restrição de espaço, maiores detalhes sobre a linguagem são omitidos sem muita perda de

entendimento. Um arquivo XWizard define um formulário (elemento <wizard>) com várias telas (elemento <screen>) contendo uma seqüência de perguntas (elemento <question>) que o usuário deve responder para criar um novo documento. A resposta dada pelo autor resulta em alterações no arquétipo em termos de valores de atributos (elemento <answer>) ou na criação de novos elementos (elemento <multiple>).

Tabela 1. Estrutura da linguagem XWizard.

<i>Elementos</i>	<i>Atributos</i>	<i>Conteúdo</i>
wizard	id	screen+
screen	id, title	multiple*, question+
question	id, title, type, {path}, {file}, {itens}	answer+
multiple	id, templateType, occurs	question+, multiple*
answer	targetId, name	

A Figura 4, a seguir, ilustra um exemplo de utilização do NCLWizard. O exemplo é descrito em XWizard e é constituído por uma tela identificada como “screen0” com três perguntas identificadas por “qHtmlPane”, “qFileBox” e “qComboBox”. A correspondência entre os elementos XML e a interface é mostrada através das setas em vermelho.

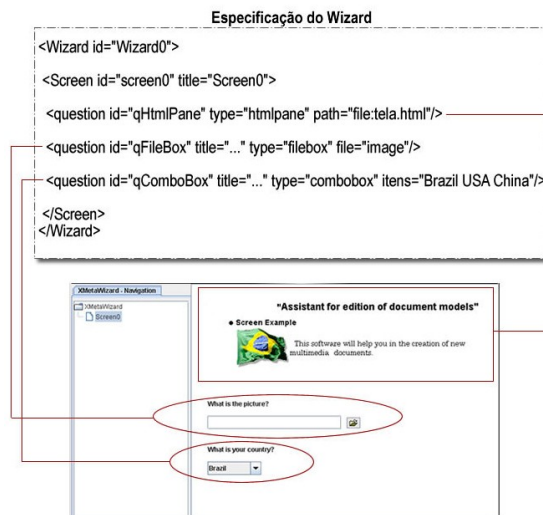


Fig. 3: Exemplo de especificação XWizard e a respectiva interface gerada.

Os arquétipos são representados e armazenados pela ferramenta NCLWizard como documentos NCL convencionais ou como *templates* NCL [15]. Suponha um arquétipo, bastante simples, que representa programas em que o espectador decide qual final, entre duas alternativas diferentes, ele deseja assistir. A estrutura do arquétipo é composta por um vídeo principal, ao final do qual é exibida uma pergunta com duas opções que permitem ao espectador decidir assistir a um dos dois vídeos finais. Essa estrutura não é alterada entre instâncias de documentos desse arquétipo. O que muda é o conteúdo dos vídeos finais e das imagens que compõem a pergunta e botões de opção. Nesse caso, é suficiente utilizar um documento NCL para

representar tal arquétipo. O autor desse arquétipo poderia ainda expressá-lo com um número variável de finais, caso em que a quantidade de vídeos e imagens de finais se tornaria variável. Para expressar arquétipos dessa natureza, a ferramenta faz uso de *templates* NCL, cuja especificação em detalhes é omitida por razões óbvias de espaço.

4 Conclusões

Sabendo-se que a quantidade de usuários finais criando *software* é bem maior que a de programadores profissionais, seja por meio de páginas *web* dinâmicas, planilhas eletrônicas, simulações científicas ou até mesmo linguagens de programação, é conveniente oferecer melhores artifícios para esse público alvo criar aplicações [12]. A proposta deste trabalho enquadra-se no âmbito do desenvolvimento, por parte desse público, de programas interativos para TV digital, explorando a autoria com base no uso de arquétipos de documentos.

Há vários indícios apontando a autoria orientada a arquétipos como bastante promissora e produtiva quando comparada a práticas convencionais de propósito geral. Isso é principalmente apontado pelas características inerentes ao ambiente de produção de conteúdo para TV. O próprio perfil profissional de produção de conteúdo contribui para a aderência da presente proposta.

O uso de arquétipos não é novo em outras áreas de autoria. Há várias abordagens no escopo de autoria multimídia para fazer o reuso de estruturas similares de documentos [8] [15]. Uma importante contribuição deste trabalho é propor que haja uma clara separação entre o ator responsável pela criação do arquétipo e aquele que exerce o papel de usá-lo na autoria de um novo documento. Mais ainda, o nível de abstração em que esses atores devem trabalhar é diferenciado visando o ganho de produtividade e simplificação para quem faz uso desses arquétipos, e não oferecendo perdas para quem os cria.

Também é proposta como prova de conceito a ferramenta NCLWizard. A ferramenta faz uso de formulários passo-a-passo para direcionar e restringir o autor na concepção de novos programas interativos. A experiência com alguns arquétipos específicos mostra que o tempo de desenvolvimento é drasticamente reduzido. O uso de testes de usabilidade para uma medição qualitativa mais acurada do ganho adquirido na autoria é proposta para trabalho futuro.

Haveria uma redução de custos significativa na concepção da base de arquétipos caso tal tarefa fosse tão simples a ponto de não requerer conhecimento especializado daquele profissional. Os *templates* NCL são uma alternativa promissora para especificar arquétipos de documentos por meio da descrição da semântica de apresentação de composições multimídia, mas ainda não são amigáveis para qualquer usuário. A simplificação do processo de autoria de *templates* NCL e possivelmente a construção de uma ferramenta visual de autoria são outra proposta de trabalho futuro.

Finalmente, sugere-se como trabalho futuro o estudo de outras abstrações ou ferramentas visuais que façam a comunicação entre o autor de arquétipos e o de documentos. Algumas possíveis sugestões são a representação por meio de grafos compostos (similar ao usado em [4]) ou o uso de planilhas eletrônicas.

Agradecimentos

Nossos agradecimentos a Bruno Seabra Nogueira Mendonça Lima e Roberto Gerson de Albuquerque pela implementação da primeira versão da ferramenta NCLWizard.

Referências

1. Antonacci M.J., Muchaluat-Saade D.C., Rodrigues R.F., Soares L.F.G.: NCL: Uma Linguagem Declarativa para Especificação de Documentos Hipermedia na Web. In: VI Simpósio Brasileiro de Sistemas Multimídia e Hipermedia - SBMídia2000, Natal, Rio Grande do Norte (Junho de 2000).
2. ARIB – Association of Radio Industries and Businesses. Data Coding and Transmission Specifications for Digital Broadcasting Volume 2: XML-Based Multimedia Coding Schema, STD-B24 Versão 4 (Fevereiro de 2004).
3. Soares, L. F. G.; Rodrigues, R. F. Nested Context Model 3.0 – Part 1: NCM Core, Monografias em Ciência da Computação, Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, 2005.
4. Coelho, R.M., Rodrigues, R.F., Soares, L.F.G.: Integração de Ferramentas Gráficas e Declarativas na Autoria de Arquiteturas Modeladas através de Grafos Compostos. In: X Simpósio Brasileiro de Sistemas Multimídia e Web, Ribeirão Preto, Brasil (Outubro de 2004).
5. W3C World-Wide Web Consortium, Synchronized Multimedia Integration Language – SMIL 2.1 Specification (Dezembro de 2005).
6. Pree, W. Framework patterns. SIGS Management Briefings. SIGS Books & Multimedia (1996).
7. ECMA. Standardizing Information and Communication Systems. ECMAScript Language Specification, Padrão ECMA 262, 3a Edição (1999).
8. Celentano, A., Gaggi, O., Template-Based Generation of Multimedia Presentations, Journal of Software Engineering and Knowledge Engineering, v. 13, n. 4, p. 419-445 (2003).
9. Gama, E., Helm, R., Jonhson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1ª Edição (Janeiro 1995).
10. Bass, L., Clements, P., Kazman, R. Software Architectures in Practice. Addison-Wesley, 2a. Edição (Abril de 2003).
11. Kiczales, G., et al. Aspect-Oriented Programming. In proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finlândia. Springer-Verlag LNCS 1241 (Juho de 1997).
12. Lieberman, H., Paternó, F., Klann, M., Wulf, V. End-User Development: An Emergin Paradigm. In End-User Development (Lieberman, Paternò and Wulf, Eds.). Springer (2005).
13. ETSI – European Telecommunications Standards Institute. Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1, Especificação Técnica ETSI TS 102 B12 (Maio de 2005).
14. van Welie, M., van der Veer, G. & Eliens, A. Patterns as Tools for User Interface Design. International Workshop on Tools for Working with Guidelines, Biarritz, França, pp. 313–324 (2000).
15. Silva, H.V.O, Muchaluat-Saade, D.C, Rodrigues, R.F., Soares, L.F.G., NCL 2.0: Integrating New Concepts to XML Modular Languages, The ACM Symposium on Document Engineering 2004, p. 188-197 (2004).