

## 5 Método de Autoria Orientado a Templates

Este capítulo propõe um método que visa fornecer um conjunto de regras básicas regendo um processo de autoria de documentos que possibilita a autores com pouca ou nenhuma experiência em programação criarem documentos hipermídia baseados em templates previamente definidos por outros autores, esses mais especializados. A tarefa do autor menos especializado é resumida a preencher as lacunas que tornam seu documento único com relação à família de documentos caracterizada por aquele template em uso. A Seção 5.1 descreve o método. Na Seção 5.2 é feita uma breve discussão sobre o emprego do método na área de concepção de aplicações interativas para TV digital. A Seção 5.3 descreve um caso de uso do método baseado em ferramentas *wizard*.

### 5.1. Método de Autoria Orientado a Templates

O método proposto envolve dois atores principais: i) o *autor de templates*, que é mais especializado e responsável pela identificação e concepção dos templates; e ii) o *autor de documentos*, que tem uma base de conhecimento menos especializada e precisa apenas entender superficialmente o template e como preencher suas lacunas. O presente método visa facilitar o trabalho do *autor de documentos* quando uma mesma estrutura recorrente de apresentação é identificada. Isso é feito sem impor a ele pré-requisitos especializados de programação. A principal característica do método é isolar o trabalho desses dois atores em níveis de abstração diferentes. Por outro lado, idealmente, busca-se diminuir a distância cognitiva entre os dois atores, tornando o autor de documentos um possível autor de templates em outras aplicações do método.

O *autor de templates* pode ter um perfil mais especializado. Cabe ao *autor de templates* a tarefa de identificar famílias de documentos e representá-las em uma linguagem de especificação voltada para esse fim. É bem-vindo que a tarefa desse autor seja facilitada por um ambiente de autoria próprio, mas esse não é o

foco principal do método. O importante para o método é que seja possível comunicar ao *autor de documentos* o significado do template até certo nível em que ele saiba instanciá-lo. Isso garante o reúso da especificação de famílias de documentos e o isolamento de ambos os autores em seu próprio nível de abstração e tarefa.

O *autor de documentos*, por sua vez, instancia *templates de documentos*. O resultado final de sua tarefa é a criação de documentos hipermídia em uma linguagem de autoria alvo. Para chegar a esse fim, no entanto, duas etapas são sequencialmente executadas. Em uma primeira etapa, o autor de documentos busca em uma base de dados um template que atende a suas necessidades e que pretende instanciar. Em uma segunda etapa, ele identifica as lacunas daquele template, as quais devem ser preenchidas para gerar um documento final. Essa especificação do conteúdo dessas lacunas do template forma o chamado *documento de preenchimento* e é feita geralmente usando uma especificação incompleta em uma linguagem para autoria de documentos hipermídia.

A Figura 28 descreve diagramaticamente o método. O fluxo de trabalho do *autor de templates* passa pelo emprego de um *ambiente de autoria para templates*. Esse ambiente fornece abstrações gráficas ou visuais para facilitar seu propósito de gerar *templates de documentos* em uma linguagem voltada para essa finalidade, como exemplificado por TAL.

O fluxo de trabalho do *autor de documentos* também passa pelo uso de um *ambiente de autoria* próprio. Esse ambiente deve fornecer facilidades para comunicar o significado dos templates disponíveis para o autor, o que visa apoiar a primeira etapa de sua tarefa, que é identificar um template que possa atender a suas necessidades. Uma vez que o template seja escolhido, o ambiente também deve fornecer facilidades para a segunda etapa de sua tarefa, que é a instanciação do template, o que é representado na Figura 28 por uma linguagem para instanciação de templates. O resultado direto do trabalho do *autor de documentos* é um *documento de preenchimento*, que na verdade trata-se de um documento hipermídia incompleto, fornecendo a informação sobre como preencher as lacunas definidas pelo template em uso.

Ainda sobre o *ambiente de autoria* para o *autor de documentos*, o significado do template pode ser abstraído por uma linguagem visual que permita isolar os dois tipos de autores em níveis de abstração diferentes. Isso é

particularmente importante, pois a existência dessa linguagem possibilita menor experiência por parte do autor de documentos, que passa a não precisar necessariamente entender a linguagem em que o template foi especificado. Outro aspecto importante é que o *autor* de documentos só precisa especificar as lacunas que tornam aquele documento único em sua família, o que não apenas facilita sua tarefa como também a torna mais rápida e menos propensa a erros.

O fluxo de trabalho encerra pela geração de um *documento hipermídia* completo em uma linguagem de autoria alvo, como NCL, SMIL ou HTML. Isso é obtido como saída de um *processador de templates*. Esse processador recebe como entrada o template que está sendo utilizado e o *documento de preenchimento*, os quais juntos permitem instanciar o template como um documento completo, pronto para exibição.

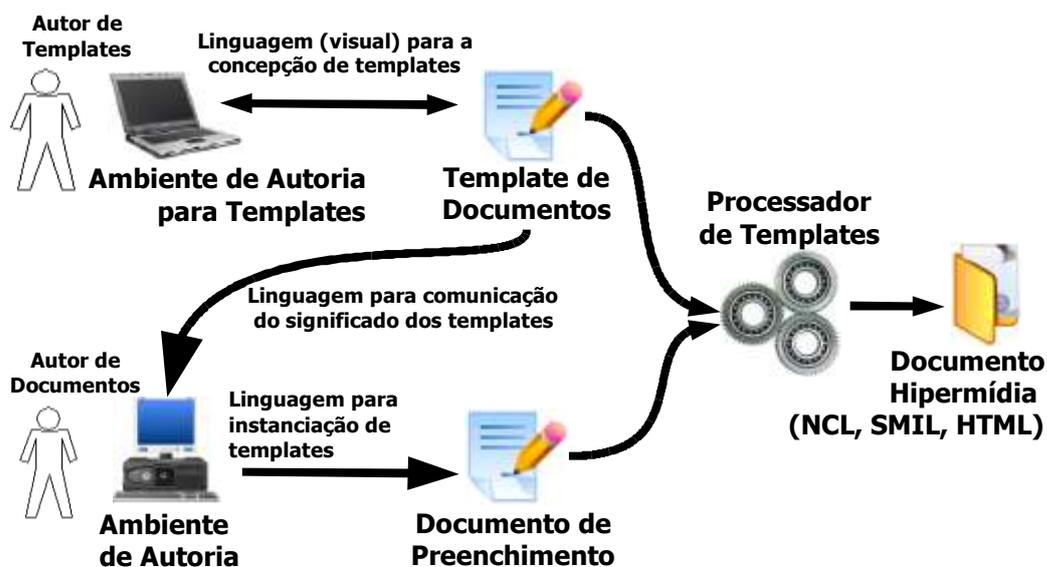


Figura 28. Autoria de Documentos Hipermídia Orientada a Templates.

É conveniente reforçar que o fluxo de trabalho do *autor de documentos* tem bem mais importância do que o do *autor de templates*. Isso porque o interesse principal do método está mais direcionado em facilitar a instanciação por parte do autor menos especializado e não em prover facilidades ao autor especializado. Isso se justifica também por se esperar a existência de muito mais autores de documentos que autores de templates.

Retomando o processo de autoria de documentos hipermídia de forma geral, é conveniente salientar que ele envolve tanto a *concepção* quanto a

*implementação* da aplicação resultante. A *concepção* é a fase mais criativa e de planejamento no processo de construção da aplicação. Pela *implementação* entende-se a codificação da aplicação na linguagem de autoria alvo. É importante que um ambiente de autoria forneça facilidades para seu usuário (e autor) tanto na concepção do documento – ajudando o autor a ser mais criativo - quanto em sua implementação – acelerando, simplificando e tornando mais eficiente a codificação. O método de autoria hipermídia orientado a templates atua fornecendo facilidades tanto na *concepção* quanto na *codificação*. Na concepção, o autor pode fazer uso de especificações incompletas, podendo trabalhar por analogia, o que seguramente facilita a criação de novos documentos baseados na estrutura de outros documentos. Na codificação, como o autor apenas precisa especificar partes de um documento incompleto, sua tarefa de implementação é reduzida e facilitada.

Não só o método proposto, mas o próprio emprego de composições como uma abstração central do processo de criação de documentos parece ser uma escolha natural para a *concepção* do documento, apoiando o autor a fazer a decomposição de sua ideia geral em níveis cada vez maiores de detalhes. Adicionalmente, é direto perceber como o emprego de composições apoia também na *codificação* do documento, tornando mais fácil a criação de documentos estruturados.

A linguagem TAL, apresentada no Capítulo 4, foi concebida para aderir bem ao método de autoria orientado a templates. Um objetivo colateral alcançado com TAL é permitir especificar a semântica de uma composição de forma explícita. Ao definir um template como uma composição hipermídia em aberto, novas composições podem ser criadas especializando essa especificação. Visto dessa forma, um template pode ser entendido como a descrição de uma semântica de relacionamentos de uma composição, a qual pode ser aplicada livremente para se criar novas composições.

TAL faz uso de conceitos conhecidos pelo autor de documentos para a modelagem de templates de documentos. Sempre usando da terminologia do método, essa decisão de projeto assegura uma forma natural (ou pelo menos simplificada) de fazer a comunicação do significado do template ao autor de documentos. Isso porque tanto no template quanto no documento hipermídia se

empregam conceitos similares, sem a necessidade de notações externas a esse domínio de problema.

## 5.2.

### **Aplicações interativas para TV digital: cenário de uso do método de autoria orientado a templates**

No ambiente de TV digital, diversas famílias de documentos podem ser identificadas na concepção de aplicações interativas. Ao se analisar um conjunto considerável e aleatório de aplicações para TV digital oriundos de vários autores, é comum verificar a repetição de uma mesma estrutura de apresentação temporal ou espacial. Alguns exemplos são menus de opções com os botões coloridos do controle remoto, a ativação da interatividade no início das aplicações ser marcada por um ícone de interatividade etc.

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 inicialmente 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 de aplicações para TV digital em famílias de documentos 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 em vez de criar o documento por inteiro. Informalmente, uma família pode ser vista como um documento quase pronto que ainda requer o preenchimento de alguns pontos em aberto que os tornam único.

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 haver na TV 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 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 telespectador ao fácil reconhecimento do funcionamento de um novo programa. 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 ainda mais a conveniência de se poder especificar famílias de documentos.

Conforme caracterizado e justificado não apenas nessa seção, o presente método de autoria orientado a templates parece aderir bem às necessidades de um ambiente de desenvolvimento de aplicações interativas para TV digital, onde é comum a existência de programas que repetem um mesmo formato de aplicação.

### 5.3.

#### **NCLWizard: caso de uso do método de autoria orientado a templates baseado em ferramentas wizard**

Um *wizard* (assistente) 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 eletrônicos, e ainda na automatização do processo de geração de código, sempre que é detectado um processo repetitivo em uma implementação (van Welie *et al.*, 2000). 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*.

O NCL Wizard é um caso de uso do método de autoria hipermídia orientado a templates. Fazendo uso da nomenclatura proposta no método, artefatos *wizards* são empregados como *ambiente de autoria* para o *autor de documentos*. 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 template específico. O usuário de um *wizard*, por sua vez, faz o papel de *autor de documentos*, focalizando apenas no preenchimento das lacunas requisitadas passo-a-passo para aquele template. O formato original do template 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 templates. O *NCLWizard* é composto por três elementos principais (vide Figura 29): 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 template de acordo com as opções e respostas do usuário.

A criação de um novo *wizard* está sempre associada a um novo template. Esse template 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. 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 template; depois, cria-se a interface do *wizard* e são especificadas as transformações que devem ser aplicadas ao template de acordo com as respostas do usuário, para se

gerar o novo documento. A primeira fase é concebida pelo *autor de templates*, conforme descrito pelo método empregado. A novidade é que na segunda fase entra em cena o Criador de Wizards (geralmente o próprio *autor de templates*), que utiliza como base o template 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 29.

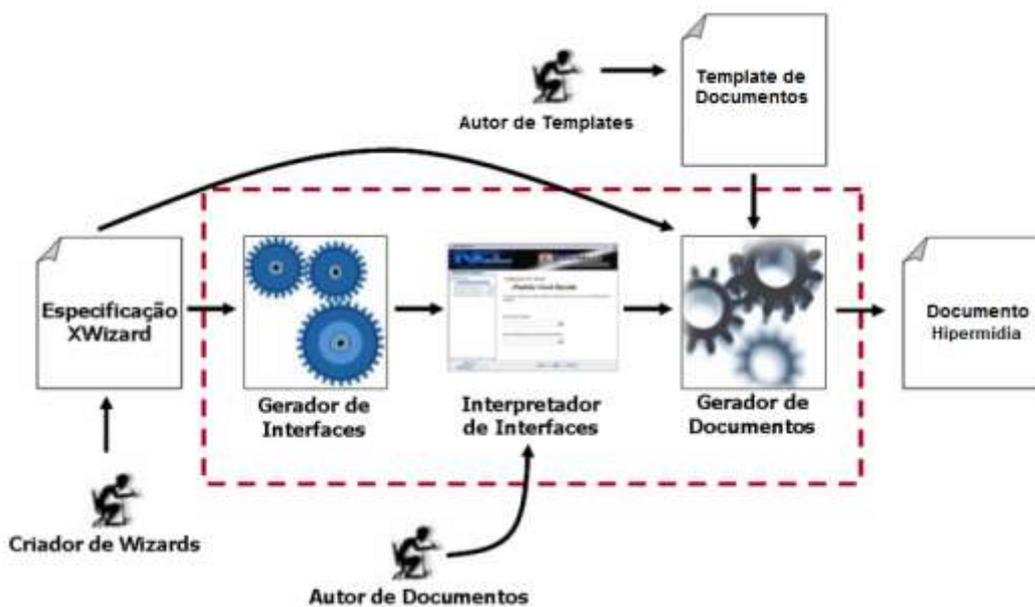


Figura 29. Atores e Arquitetura da ferramenta NCLWizard.

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 template 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 para gerar um novo documento com base nas respostas do autor.

A Tabela 4 lista os elementos e atributos de *XWizard*. Maiores detalhes sobre a linguagem são omitidos sem prejuízo 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 template em termos de valores de atributos (elemento <answer>) ou na criação de novos elementos (elemento <multiple>).

Tabela 4. Estrutura da linguagem XWizard.

<b>Elemento</b>	<b>Atributos</b>	<b>Conteúdo</b>
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.



Figura 30. Exemplo de especificação XWizard e a respectiva interface gerada.

É conveniente ressaltar, com o propósito de evitar confusão de termos, que os templates são representados e armazenados pela ferramenta NCLWizard como documentos NCL convencionais ou como documentos TAL. Suponha um template, bastante simples, que representa programas em que o espectador decide qual final, entre duas alternativas diferentes, ele deseja assistir. A estrutura do template 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 template. 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 o template. O autor desse template 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 templates dessa natureza dinâmica (a maioria deles), a ferramenta faz uso de especificações em TAL.