

### 3 Trabalhos Relacionados

A *Web* e alguns sistemas de TV Digital podem ser destacados como exemplos de ambientes que dão suporte ao desenvolvimento de aplicações hipermídia dinâmicas. Em geral, esse dinamismo é possibilitado por mecanismos que se baseiam em linguagens imperativas (de *script*), que oferecem funcionalidades para manipulação da estrutura dos documentos. A complexidade da autoria é impactada, pois há a necessidade não somente do desenvolvimento de um *script* como também de se elaborar um documento hipermídia susceptível à manipulação imperativa.

Considerando esses dois ambientes de execução de aplicações hipermídia (TV Digital e *Web*), este capítulo analisa alguns trabalhos relacionados que possuem características relevantes para o desenvolvimento desta dissertação. Os trabalhos apresentados dão suporte ao desenvolvimento de aplicações dinâmicas - entre as quais estão aquelas que se recriam dinamicamente - nos ambientes mencionados.

Quando uma página *Web* é criada utilizando somente HTML, todo o conteúdo dessa página é determinado pelo autor da página. Qualquer visitante visualiza um mesmo conteúdo, pois a página *Web* é definida por um documento hipermídia estático. Por outro lado, o conteúdo de uma página *Web* dita dinâmica pode sofrer mudanças em tempo de exibição. Por exemplo, páginas de notícias que apresentam suas manchetes mais recentemente adicionadas ao banco de dados utilizado para construir e atualizar essa página. Quando um novo item é adicionado ao banco de dados, e uma nova requisição a esta página é solicitada, o conteúdo da página em questão passa a exibir as notícias mais recentes.

Conforme já discutido no capítulo anterior, o dinamismo pode ser suportado tanto pelo lado do cliente quanto pelo lado do servidor. Se o *script* for executado do lado do servidor, a página só é atualizada quando uma nova requisição é realizada explicitamente. Por outro lado, ao executar o *script* também no lado do cliente, tal requisição pode ser periódica, permitindo que a página seja

dinamicamente modificada, sem solicitação direta por parte do usuário, ou seja, a página se recria dinamicamente em tempo de apresentação. No ambiente de TV Digital, onde temos uma comunicação do tipo *push*, não há a necessidade de uma requisição periódica, visto que sempre que novos dados estejam disponíveis eles serão enviados, sem solicitação do usuário.

Nos dois ambientes a autoria de documentos dinâmicos envolve maior complexidade em relação aos documentos estáticos devido aos elementos imperativos inseridos – como os *scripts* ou *applets* Java mencionados anteriormente – para acrescentar dinamismo nesses documentos. Como mencionado anteriormente, o uso de *templates* para possibilitar a criação de documentos dinâmicos permite que seja possível manter o dinamismo a partir de uma abstração de autoria completamente declarativa (o que a princípio é mais simples).

A aplicação cuja arquitetura é especificada neste trabalho pode ser compreendida como uma espécie de gerador automático de *mashups* (documentos hipermídia que usam conteúdo de mais de uma fonte para criar um novo serviço completo). Essa associação reside no fato de que, conforme descrito na Seção 4.1, tal aplicação provê uma série de opções de conteúdos oriundos de diferentes fontes, que ao serem selecionados, ocasionam na geração automática de uma nova especificação da aplicação. Em (Wong, 2007) é proposto um sistema para permitir que o usuário final de uma aplicação *Web* possa compor *mashups* a partir de conteúdo e serviços disponibilizados na *Web*, partindo de um conjunto de abstrações visuais e operadores (similares aos *pipes* oferecidos por *Unix*) que são utilizados para definir um diagrama de fluxo de dados, gerando assim um novo serviço *Web*.

A proposta desta dissertação está relacionada com o que é feito por algumas ferramentas que criam *mashups* em ambientes de execução hipermídia. A ferramenta apresentada em (Hartmann, 2007), por exemplo, tem o objetivo de criar *mashups* para *Web*, a partir do uso de *templates*. Parte da contribuição desta dissertação é a implantação dessa abordagem dentro de um ambiente de TVD.

No âmbito da TV Digital, algumas ferramentas de autoria fazem uso de *templates* (Chiao, 2006) para facilitar o desenvolvimento de aplicações interativas. Ferramentas comerciais como a Cardinal Studio Professional 4 (Cardinal Systems, 2004a) (Cardinal Systems, 2004b) dão suporte à utilização de definições que

generalizam o comportamento de alguns tipos de aplicações interativas MHP, de forma que o desenvolvimento de aplicações que se enquadrem nos tipos previstos seja simplificado. Em (Chiao, 2006) é proposta uma ferramenta de autoria baseada em *templates* XML, que descreve em alto nível o comportamento espaço-temporal da aplicação interativa. A descrição é usada para geração de código Java, de forma que possa ser executada em receptores compatíveis com o *middleware* MHP. Comumente, ferramentas de autoria de aplicações interativas para TV Digital não facilitam a construção de uma aplicação dinamicamente recriável. Tais ferramentas utilizam *templates* para facilitar o reuso de estruturas de aplicações interativas, eliminando alguns passos para os desenvolvedores de aplicações, e não usuários de aplicações de TV Digital interativa.

O trabalho desenvolvido em (Lima, 2009) utiliza NCL e tem como foco o ambiente de TV Digital. Ele propõe mecanismos para auxiliar na autoria de novas aplicações por meio de *templates*. Nesse trabalho, os *templates* são documentos incompletos compostos por lacunas, que são partes editáveis passíveis de modificação pelo autor de uma nova aplicação. É proposto um esquema declarativo de anotações em documentos multimídia que identifica estruturas recorrentes e facilita o preenchimento das lacunas. Um processador de *templates* foi implementado de forma a permitir a geração de uma instância de um *template* baseado em dados fornecidos pelo autor.

Ainda no que se refere à TV Digital, outro trabalho que é relacionado a esta dissertação é o apresentado por (Soares Neto, 2008), onde é definido o NCL Wizard. A abordagem de (Soares Neto, 2008) também é voltada para a utilização de *templates* em TV Digital, mais precisamente para NCL. NCL Wizard é uma ferramenta que se utiliza de *XTemplate* (Saade, 2003) e é utilizado apenas em tempo de autoria, não oferecendo, portanto, a possibilidade de recriar a aplicação em tempo de apresentação. Uma relação que é possível estabelecer o presente trabalho e NCL Wizard seria a exploração dessa ferramenta, em tempo de autoria, para a geração do documento inicial.

Abordagens mais genéricas também podem ser encontradas na literatura, como é o caso do proposto em (Celentano, 2003). Esse trabalho apresenta uma solução para a geração automática de apresentações multimídia, dividida em duas fases. A primeira fase é a definição do *template*. Na segunda fase, que deve ser executada para cada nova apresentação, é realizada a busca do conteúdo dinâmico,

conforme especificado pelo *template*, que é preenchido. Diferentemente do que é proposto por (Celentano, 2003), a arquitetura do presente trabalho oferece a possibilidade de recriação da aplicação – a segunda fase proposta por (Celentano, 2003) – a partir do ambiente de execução do cliente.

Por fim, é importante ressaltar que o método de autoria orientado a *templates* desenvolvido em (Soares Neto, 2010), descrito anteriormente no Capítulo 2, foi utilizado como diretriz durante a especificação da arquitetura proposta nesta dissertação. O método de autoria de (Soares Neto, 2010) foi adaptado para as características das aplicações que se recriam dinamicamente. Os documentos finais obtidos a partir do método em (Soares Neto, 2010) estão prontos para serem exibidos e não prevêm novos ciclos de autoria, diferentemente das aplicações que se recriam dinamicamente tratadas no escopo desta dissertação, nas quais interações do usuário iniciam a recriação dos documentos a partir de um *template*.