

1

Introdução

Em sua origem, a WWW - *World-Wide Web* (Berners-Lee, 1994) foi concebida como uma aplicação de hipertexto, visando apresentar informações científicas com referências cruzadas e permitindo a pesquisa automática de textos. Para especificação de documentos nessa aplicação, foi definida a primeira versão da linguagem HTML, que veio a se tornar um padrão do *W3C Consortium*¹. Universalmente adotada nos dias atuais, HTML (W3C, 1999a) é a principal linguagem para a autoria de documentos na rede mundial de computadores.

O fenômeno de difusão da Internet ultrapassou todas as previsões e o paradigma de hipertexto, utilizado pela linguagem HTML em sua forma original, evoluiu para o paradigma hipermídia. Proporcionando a incorporação de novos formatos de mídia, o paradigma hipermídia traz também novos requisitos, como a definição de sincronismo temporal e espacial entre componentes de um documento. Tais requisitos vêm motivando uma extensa pesquisa na área de sistemas hipermídia, acarretando a definição de novos padrões, tanto para a WWW como para outros sistemas.

Contextualizando o trabalho, este capítulo apresenta, inicialmente, uma visão geral de sistemas hipermídia e conceitos relacionados. Baseados nesses conceitos, serão apontados a motivação e os objetivos desta dissertação. Finalmente, a organização do restante do texto é apresentada.

1.1.

Motivação

Um sistema hipermídia possui, tipicamente, os seguintes subsistemas²: o de autoria (edição), o de armazenamento e o de execução (exibição). Durante o

¹ <http://www.w3c.org/>

² Os *subsistemas* também podem ser chamados de *ambientes* ou *módulos* do sistema hipermídia.

processo de autoria, o sistema deve oferecer ferramentas que possibilitem especificar um documento hipermídia de acordo com as concepções do autor. Essa especificação pode, então, tanto ser armazenada quanto entregue para exibição. A Figura 1:1 ilustra os subsistemas de um sistema hipermídia.

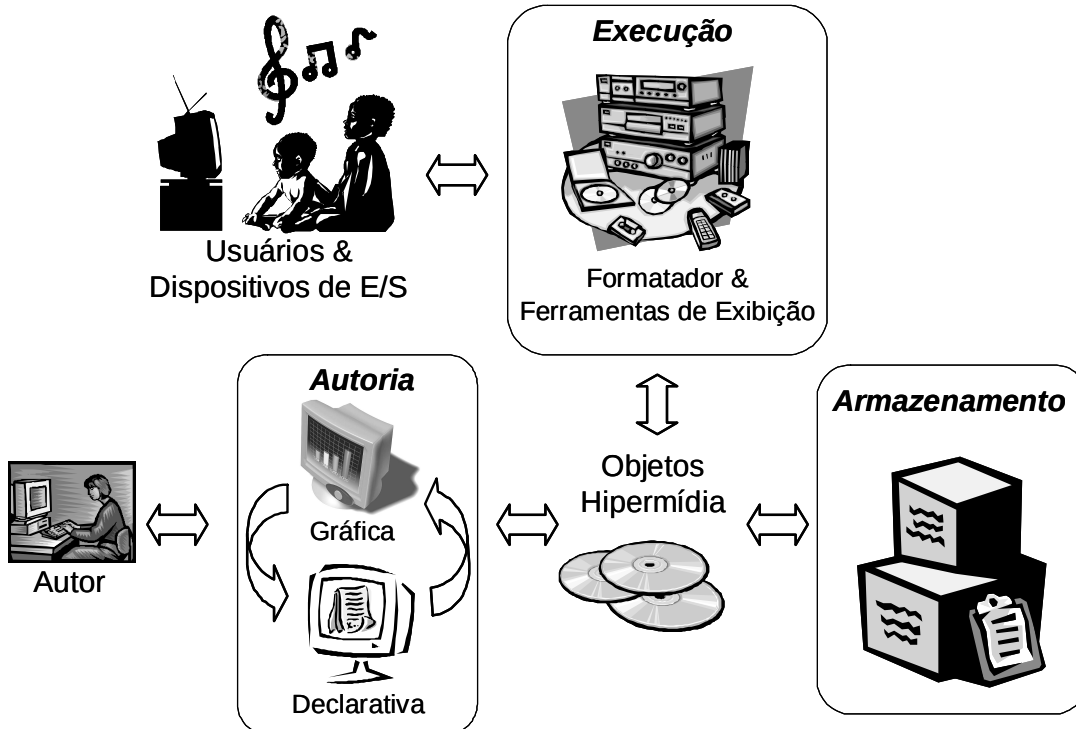


Figura 1:1. Subsistemas de um sistema hipermídia

O subsistema de autoria deve fornecer os recursos necessários para que um autor (ou autores) especifique um documento hipermídia. Nesse subsistema, é desejável a presença de ferramentas de autoria seguindo tanto o paradigma gráfico (baseado em interfaces gráficas) quanto o paradigma declarativo (baseado em linguagens textuais) - como apresentado em (Coelho, 2004).

O subsistema de armazenamento é responsável pelo armazenamento, recuperação e, em certos casos, adaptação de documentos³. Em geral, os dados armazenados são classificados em *descrição dos documentos* (contendo informações a respeito da estrutura do documento, os relacionamentos entre seus componentes e especificações de apresentação) e *conteúdo dos objetos de mídia* (arquivos de texto, imagem, áudio, vídeo etc. referenciados pela descrição dos documentos). Um dos formatos possíveis para o armazenamento das descrições de documentos é a especificação em uma linguagem declarativa.

Finalmente, o subsistema de execução (ou de apresentação) contém os mecanismos para interagir com os usuários e controlar a exibição dos objetos de mídia, de forma a respeitar as especificações do documento. Entre os elementos que compõem o ambiente de execução, podem ser destacados: as ferramentas de exibição e o formatador hipermídia. As ferramentas de exibição são compostas pelos recursos de hardware ou software responsáveis por tratar os dispositivos de entrada e saída (*Dispositivos de E/S* na Figura 1:1), controlando a exibição do conteúdo dos objetos e a interação com o(s) usuário(s). O formatador hipermídia é o nome dado ao elemento que reúne as funções para o controle da apresentação. Para realizar suas tarefas, o formatador deve receber a especificação de um documento que, como já mencionado, pode ser baseada em uma linguagem declarativa.

A adaptação de documentos é uma funcionalidade desejável em sistemas hipermídia. Seja no subsistema de armazenamento ou no de execução, a adaptação de documentos pode atuar sobre a estrutura, apresentação ou o próprio conteúdo dos objetos. Essa adaptação pode ser direcionada pelas características da plataforma de armazenamento (capacidade de processamento disponível, banda passante disponível para o acesso aos recursos gravados em memória secundária, número de clientes conectados, banda passante para transmissão dos dados no sistema de comunicação etc.) e, também, pelas informações provenientes do ambiente de execução (preferências e conhecimentos do usuário, recursos de hardware disponíveis na plataforma de exibição, localização etc.). Todas essas informações fazem parte do contexto de apresentação do documento (Boll & Klas, 1999; Brusilovsky, 1996; Dey et al., 2001; Schilit et al., 1994; Villard et al., 2000). Dessa forma, é desejável que linguagens declarativas hipermídia ofereçam suporte à criação de documentos adaptáveis.

Esta dissertação tem como foco linguagens para autoria declarativa de documentos hipermídia. A expressividade dessas linguagens está diretamente relacionada com o modelo hipermídia no qual elas se baseiam. Analisando os diversos modelos hipermídia propostos na literatura, podem ser observadas duas entidades básicas: nós (representando componentes de conteúdo de um documento) e elos (representando relacionamentos entre nós) (Muchaluat-Saade,

³ O conceito de adaptação será detalhado posteriormente.

2003). Frequentemente, uma terceira entidade também é encontrada, chamada nó de composição, ou simplesmente composição. Assim como elos, nós de composição são utilizados para representar relacionamentos entre componentes de um documento. Visando um aumento de expressividade (dentre outros benefícios), modelos mais complexos possuem outras entidades. A complexidade e o poder de expressão dos modelos são, obviamente, herdados por suas linguagens.

Com o surgimento do padrão XML (W3C, 2000b), linguagens declarativas, nele baseadas (aplicações XML), foram definidas para diferentes modelos hipermídia. O modelo hipermídia CMIF (van-Rossum et al., 1993), por exemplo, serviu como base para a linguagem *Synchronized Multimedia Integration Language* - SMIL (W3C, 2001b), enquanto o modelo *Nested Context Model* - NCM (Soares et al., 2003) originou a linguagem *Nested Context Language* - NCL (Muchaluat-Saade et al., 2003; Muchaluat-Saade, 2003).

Comparando as diversas linguagens hipermídia existentes, podem ser observadas características e funcionalidades presentes somente em algumas delas. Isso ocorre com NCL, por exemplo, que possui o conceito de conectores hipermídia e de *templates* de composição - conceitos não encontrados em outras linguagens hipermídia. Conectores hipermídia (definidos pela linguagem XConnector de NCL) são entidades que expressam a semântica de uma relação, podendo ser reutilizados para criação de diferentes relacionamentos (elos). *Templates* de composição (definidos pela linguagem XTemplate de NCL⁴), por sua vez, especificam a semântica de composições (semântica temporal paralela, semântica temporal seqüencial etc.), podendo ser reaproveitados por diferentes composições, que herdam as especificações definidas no *template*.

Diferente de NCL, composições SMIL possuem uma semântica temporal pré-definida (seqüencial, paralela etc.). Em diversos casos, as composições com semântica temporal em SMIL oferecem uma grande facilidade para autoria de documentos. Entretanto, a facilidade de autoria oferecida está restrita ao uso das composições previamente definidas pela linguagem. Além das composições, um número limitado de eventos podem ser utilizados para definição de elos

⁴ As linguagens XConnector e XTemplate fazem parte do conjunto de especificações da linguagem NCL.

(relacionamentos) entre nós. Porém, na especificação de relacionamentos complexos, é necessária a combinação de diversos eventos e/ou composições SMIL. Essa abordagem dificulta o processo de autoria e, principalmente, o reuso.

Visando aumentar o reuso e facilitar ainda mais o processo de autoria, é interessante incorporar o conceito de conectores e *templates* hipermídia à linguagem SMIL. Conectores podem definir relações simples ou complexas, e permitem que qualquer relacionamento SMIL (tanto via composições, como via eventos) seja declarado da mesma maneira: através do reuso de conectores. O uso de *templates* de composição possibilita a definição de diferentes semânticas para composições, criando, dessa forma, um mecanismo para tornar SMIL extensível em relação às composições que oferece.

Este trabalho engloba não apenas a expressividade e facilidade de reuso em linguagens hipermídia (notadamente: NCL e SMIL), mas também a conversão (interoperabilidade) entre os diversos formatos abstratos por elas gerados. Assim, por exemplo, um documento SMIL pode ser editado em um ambiente de autoria utilizando-se de um outro formato de dados (como objetos Java de uma implementação do modelo NCM). De forma análoga, para que esse documento seja enviado para o subsistema de armazenamento, pode ser necessária sua conversão para um outro formato. Finalmente, para sua apresentação, pode ser ainda exigida uma conversão para o formato de dados do formatador hipermídia que irá exibi-lo. Essas conversões são realizadas por compiladores de documentos de linguagens hipermídia, como será apresentado.

1.2. Objetivos

Os dois principais objetivos desta dissertação são: (1) aumentar o reuso e a expressividade da linguagem SMIL, especificando a linguagem X-SMIL; e (2) definir *frameworks* de compiladores e implementar suas instâncias (compiladores específicos) para processar documentos nessas linguagens. Como decorrência desses objetivos, um terceiro surge naturalmente para viabilizá-los: (3) o refinamento da linguagem NCL, introduzindo a versão 2.1 dessa linguagem - que estende sua versão anterior (2.0).

Assim, inicialmente, o refinamento à linguagem NCL será proposto. Entre as principais melhoras destacam-se a definição de um módulo⁵ NCL para a especificação de funções de custo para as durações de objetos de mídia (Rodrigues, 2003) e a definição de um módulo para a especificação de regras de apresentação, buscando auxiliar a autoria de documentos adaptativos. A maior contribuição nesse refinamento, no entanto, é a redefinição da linguagem XTemplate de NCL, utilizada para a especificação de *templates* de composição.

A proposta para a extensão de SMIL será desenvolvida introduzindo, nessa linguagem, o conceito de *templates* de composição e o conceito de conectores. Dessa forma, uma nova linguagem, chamada X-SMIL, é especificada. Em X-SMIL serão permitidos: a definição e reuso de especificações semânticas para composições - através de *templates* de composição; e a definição e reuso de especificações de relações - através de conectores (que podem ser entendidos, nesse contexto, como *templates* para elos).

Na definição da linguagem X-SMIL, adotou-se uma estratégia para incorporação dos conceitos mais importantes de NCL. Assim, a linguagem X-SMIL é definida por meio de duas extensões à linguagem SMIL: o perfil⁶ SMIL + XTemplate (ou XT-SMIL), que introduz as diversas vantagens do uso de *templates* de composição em SMIL; e o perfil SMIL + XConnector (ou XC-SMIL), que incorpora o conceito de conectores (e a definição de elos a partir de referências a conectores) à linguagem SMIL.

Apesar de a linguagem XTemplate de NCL permitir relacionamentos outros que não apenas aqueles definidos por conectores, a versão XTemplate de NCL 2.0 (ou, simplesmente, XTemplate 2.0) era limitada a esses relacionamentos. Como, a princípio, no perfil XT-SMIL não seria viável a utilização de *templates* de composição sem o uso de conectores, foi preciso redefinir a linguagem XTemplate 2.0, dando origem à linguagem XTemplate 2.1. Essa redefinição consiste na sua estruturação em perfis, dependentes das relações para as quais provêem suporte. Um desses perfis, independente de conectores e que possibilita a

⁵ O conceito de módulos de linguagens será apresentado no Capítulo 2.

⁶ Perfis de linguagem reúnem um subconjunto dos módulos oferecidos pela linguagem, definindo assim um subconjunto de funcionalidades apropriadas para a construção de uma determinada classe de documentos.

definição de relações de inclusão, é o utilizado em XT-SMIL. Relações de inclusão⁷ constituem a forma mais simples de se introduzir o conceito de *templates* em SMIL.

A definição do perfil XC-SMIL é simples, pois XConnector é um módulo independente em NCL. Finalmente, é definida a linguagem X-SMIL, resultante da combinação de XT-SMIL e XC-SMIL. Em X-SMIL, o perfil de XTemplate 2.1 utilizado pode definir relações através de conectores ou relações de inclusão. Esse perfil de XTemplate é, exatamente, o definido pela linguagem NCL 2.1.

Como mencionado, um dos objetivos deste trabalho é a análise da interoperabilidade entre as linguagens NCL, SMIL e X-SMIL - através de compiladores de documentos de cada uma dessas linguagens para documentos nas outras duas. A compilação de documentos nessas três linguagens para uma implementação do modelo NCM em Java e para o modelo de execução do Formatador HyperProp (Rodrigues, 2003) também será abordada, com o objetivo de visualização gráfica, edição e execução (exibição) desses documentos no sistema HyperProp (Soares et al., 2000).

Como cada grupo de compiladores (de NCL, de SMIL e de X-SMIL) analisa documentos de uma mesma linguagem de origem, várias funcionalidades podem ser reaproveitadas e implementadas uma única vez. A identificação dessas funcionalidades constitui o primeiro passo de implementação, dando origem à definição de *frameworks* para compiladores, como o *framework* para compiladores NCL.

Além das características comuns aos compiladores de uma mesma linguagem, observou-se, em outro nível de abstração, aspectos comuns que podem ser reutilizados em processadores de qualquer linguagem baseada em XML. A partir dessas observações, uma estrutura em dois níveis para especificação de *frameworks* para implementação de compiladores é proposta. Em um primeiro nível, existe o *framework genérico de processamento* (ou *meta-framework*), cujo objetivo é facilitar o desenvolvimento de *frameworks* de compiladores. O *meta-framework* define, sintática e semanticamente, os métodos de *frameworks* de compiladores e a estruturação das classes desses *frameworks* (que são gerados

⁷ Relações de inclusão permitem especificar as composições nas quais determinados objetos devem estar contidos.

automaticamente, a partir das especificações de linguagens baseadas em XML). Dessa forma, o *framework* para compiladores NCL, o *framework* para compiladores SMIL e o *framework* para compiladores X-SMIL representam instâncias do *meta-framework*.

Finalmente, é importante destacar que, juntamente com a descrição desses compiladores, será apresentada a implementação do processador de *template* para os perfis da linguagem XTemplate. Esse processador também é implementado como instância do *meta-framework* de compiladores.

1.3.

Organização da Dissertação

Esta dissertação está estruturada da seguinte forma. O Capítulo 2 apresenta linguagens para especificação de documentos hipermídia, dando destaque para as linguagens NCL e SMIL. O Capítulo 3 discute as principais funcionalidades introduzidas e refinadas na linguagem NCL. Nesse capítulo, são detalhados os novos módulos de NCL: os novos módulos para *templates* de composição (ou seja, a definição e estruturação da linguagem XTemplate 2.1), o módulo para especificação de regras de apresentação, o módulo para especificação de funções de custo e os refinamentos da linguagem XConnector. O Capítulo 4 contempla a linguagem X-SMIL, detalhando os perfis que compõem essa linguagem: XT-SMIL (SMIL + XTemplate) e XC-SMIL (SMIL + XConnector). O Capítulo 5 trata dos compiladores para linguagens XML, apresentando o *meta-framework* de compiladores, assim como os *frameworks* para compiladores instanciados a partir do *meta-framework*. O Capítulo 5 também descreve os compiladores instanciados a partir dos *frameworks* para compiladores. O Capítulo 6 traz comparações com os principais trabalhos relacionados da área e o Capítulo 7 tece as considerações finais da dissertação, salientando as contribuições do trabalho e possíveis pesquisas futuras.