

1 Introdução

Um sistema hipermídia completo deve oferecer suporte a todas as fases de tratamento de um documento, desde o momento de sua concepção pelos autores, passando pela fase de armazenamento em servidores de documentos, até que seja apresentado aos usuários finais.

Nesses sistemas, destaca-se a presença de dois componentes fundamentais que são o componente de edição (editor) e o componente de execução (executor) de documentos hipermídia. O *editor* é responsável por capturar o desejo de expressão do autor de um documento e transformá-lo em objetos hipermídia, que representam os componentes do documento e as relações entre eles. O *executor* é responsável por interpretar os objetos hipermídia, decifrando os componentes do documento e as relações por eles representados, para, em seguida, apresentá-los através de dispositivos de entrada/saída de uma plataforma de exibição. A Figura 1 ilustra a interação entre os vários componentes de um sistema hipermídia.

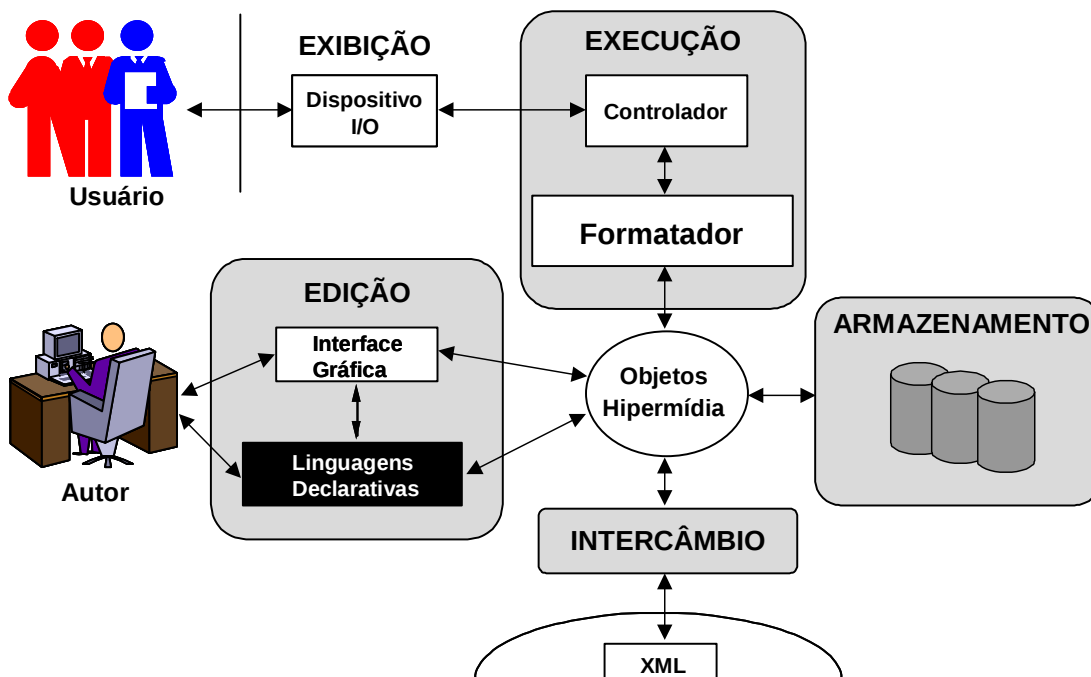


Figura 1. Componentes de um ambiente de autoria e execução de documentos hipermídia.

Além dos componentes de edição e de execução, o componente de *armazenamento* é responsável por armazenar os objetos hipermídia em bases de dados. Normalmente, os documentos hipermídia criados são armazenados em servidores hipermídia, como por exemplo servidores Web. O executor deve então conhecer o protocolo utilizado para se comunicar com os servidores, a fim de recuperar os objetos hipermídia a serem apresentados ao usuário final.

O ambiente de autoria e execução pode ter ainda um outro componente, chamado componente de *intercâmbio*, responsável por permitir a transferência de objetos hipermídia de/para outros sistemas. O padrão de fato utilizado atualmente para intercâmbio de dados entre sistemas é o formato XML (W3C, 2000), padronizado pelo W3C¹ – *World Wide Web Consortium*.

A edição de documentos pode ser feita de forma gráfica, fazendo uso de um editor que ofereça interface gráfica, ou então de forma declarativa, onde o autor utiliza uma linguagem de autoria declarativa para especificar seus documentos.

Editores gráficos, para a definição da estrutura de documentos e das relações de sincronização entre seus componentes, normalmente oferecem múltiplas visões para edição (Hardman, 1993b; Muchaluat-Saade, 1997; Jourdan, 1998). Uma das visões considera a edição da estrutura do documento, fornecendo recursos para editar componentes, seus relacionamentos, e o agrupamento desses componentes em composições. Outra visão é responsável pela especificação dos relacionamentos de sincronização temporal entre os componentes de um documento multimídia, definindo suas posições relativas no tempo. Uma terceira visão permite a definição de relacionamentos espaciais entre componentes de um documento multimídia, estabelecendo suas características de apresentação em um determinado dispositivo, definindo, por exemplo, a posição em um monitor de vídeo ou o volume em um dispositivo de áudio.

Em um editor gráfico que oferece múltiplas visões, elas devem estar relacionadas, de modo que um objeto em foco na visão estrutural seja a base para a seqüência de sincronismo mostrada na visão temporal. Nessa última visão, para cada ponto selecionado no tempo, a visão espacial mostra como os componentes

¹ <http://www.w3.org>

serão apresentados no espaço definido pelos dispositivos de saída (Muchalut-Saade, 1997).

A edição gráfica facilita bastante o processo de autoria, pois o autor consegue ter uma visão mais próxima da apresentação real do documento ao usuário final. Entretanto, ela requer o uso de editores mais elaborados e de implementação mais complexa. Uma outra alternativa para fazer a edição de documentos usa o procedimento declarativo, no qual o autor necessita apenas de um editor de texto tradicional para escrever a representação textual do documento. Nesse caso, o autor é quem cria o seu próprio mapa mental para visualizar o documento em suas diferentes dimensões, pois essa facilidade normalmente não é oferecida pelo editor. Mesmo no procedimento declarativo, o sistema de autoria deve fornecer meios para o autor integrar a representação textual do documento ao restante do sistema. Essa integração é necessária para a apresentação do documento ao usuário, além de permitir procedimentos de validação sintática e verificação semântica da especificação do documento.

Este trabalho está inserido no contexto de ambientes de autoria e execução hipermídia, sendo que seu foco principal se localiza no componente de edição responsável pelo procedimento declarativo de especificação de documentos. Alguns trabalhos anteriores discutiram a autoria gráfica de documentos hipermídia (Muchalut-Saade, 1996b; Costa, 1996b; Muchalut-Saade, 1996a; Muchalut-Saade, 1997; Pinto, 2000), mas não serão aqui abordados para simplificação deste texto.

Uma linguagem de autoria hipermídia deve satisfazer os requisitos necessários para a especificação completa de um documento (Antonacci, 2000c). Como requisito básico, a linguagem deve permitir a definição dos componentes do documento e dos relacionamentos entre eles. Além disso, pode-se identificar outros requisitos fundamentais, tais como permitir a especificação de documentos de forma estruturada, a representação de tipos diferentes de objetos de mídia, a especificação de relacionamentos espaço-temporais complexos entre componentes, a definição das características de apresentação separada da definição dos componentes, a especificação do comportamento temporal do documento de uma forma flexível, a possibilidade de adaptação de um documento

ao usuário e à plataforma em que ele estiver sendo exibido, e o reuso de especificações prontas.

Um ponto-chave para a eficácia de uma linguagem é o modelo hipermídia em que ela se baseia. Se o modelo hipermídia for muito simples, a linguagem também será e, como consequência, seu poder de expressão será pequeno. Por outro lado, se a linguagem de autoria se baseia em um modelo hipermídia mais complexo, seu poder de expressão será grande, e como consequência, sua complexidade de utilização também cresce. Pode-se afirmar que o poder de expressão de uma linguagem de autoria hipermídia é limitado pelo modelo conceitual em que ela se baseia.

Vários modelos hipermídia foram propostos na literatura (Buchanan, 1993; Vazirgiannis, 1993; Halasz, 1994; Soares, 1995; Willrich, 1996; Hardman, 1998; Jourdan, 1998; Lowe, 1999; Na, 2001; Díaz, 2001), muitos deles apresentando características distintas. Tradicionalmente, as entidades básicas encontradas em modelos conceituais hipermídia são *nós*, representando componentes de conteúdo de um documento, e *elos*, representando relacionamentos entre nós. Frequentemente, uma terceira entidade também é oferecida, chamada *nó de composição*, ou simplesmente *composição*. De forma análoga ao uso de elos, nós de composição também são utilizados para representar relacionamentos entre componentes de um documento.

Um modelo hipermídia deve dar suporte a diversos tipos de relações (Soares, 2000), entre as quais pode-se destacar:

- Relações de referência: por exemplo, aquelas que levam a uma pequena nota ou a outra seção inteiramente nova;
- Relações de estruturação: por exemplo, aquelas que especificam a estrutura lógica de um documento, tal como um livro e seus capítulos, os capítulos e suas seções etc.;
- Relações de sincronização: aquelas que definem o posicionamento temporal e espacial dos objetos;
- Relações semânticas: por exemplo, aquela entre um professor e as várias disciplinas que ele leciona ou entre um fabricante e os diversos produtos fabricados;

- Relações de derivação: por exemplo, aquelas que indicam os objetos que deram origem a outros objetos; e
- Relações de tarefas: aquelas que organizam tarefas em um trabalho cooperativo.

Diferentes tipos de elos ou de composições podem ser usados para representar os diferentes tipos de relações, sendo que nem todos os modelos conceituais dão suporte à especificação de todos os tipos. No caso de relações de referência especificamente, elos são usados tradicionalmente para representá-las em praticamente todos os modelos. No que se refere aos outros tipos de relações, o uso de elos e composições depende do modelo em questão.

Os modelos mais simples só oferecem nós e um tipo de elo, usado para representar relações de referência. Como exemplo, temos o modelo usado na *World-Wide Web* (Berners-Lee, 1994), que originou a linguagem HTML (Berners-Lee, 1995). Nesse modelo, não existe um escopo que limita a criação de relacionamentos entre nós, qualquer nó pode estar relacionado a qualquer outro nó. No caso do HTML, os elos estão definidos dentro do conteúdo do nó de origem, o que acarreta vários problemas (Rodrigues, 1998b), entre eles o de não poder reusar o nó sem herdar obrigatoriamente os elos definidos anteriormente. Outra desvantagem desses modelos é dificultar a criação de visões estruturais dos documentos, já que um diagrama estrutural mostraria inúmeros nós e elos apresentando-se totalmente ilegível (Muchaluat-Saade, 1996a). Além disso, como só oferecem um tipo de relação, seu poder de expressão é bastante pequeno.

Outros modelos oferecem nós de composição, contendo nós e elos, sem permitir o aninhamento de composições. Como exemplo, temos o sistema Intermedia (Yankelovich, 1985), que oferecia composições chamadas *webs*, para representar relações de estruturação entre nós. Uma composição contém um conjunto de nós e elos entre eles. Como elos e nós são entidades independentes, pode-se reusar o mesmo nó em composições distintas, definindo-se outro conjunto de elos para esse nó. Apesar de introduzir essa vantagem, limitando o escopo de criação de elos para nós contidos em uma composição, somente um nível de agrupamento de nós e elos não é suficiente para estruturar logicamente um documento. Por esse motivo, as outras desvantagens citadas no parágrafo anterior também se aplicam nesse caso.

Um terceiro grupo de modelos permitem que nós de composição conttenham outros nós de composição recursivamente aninhados. Como exemplo, temos o modelo Dexter (Halasz, 1994), o NCM – *Nested Context Model* (Soares, 1995) e o AHM – *Amsterdam Hypermedia Model* (Hardman, 1998), entre outros. Nesse caso, o suporte à especificação de relações de estruturação é totalmente contemplado. Como vantagens, mantém-se a possibilidade de criação de diferentes conjuntos de elos para um mesmo nó. Além disso, no caso de modelos que consideram elos como componentes de um nó de composição, como o NCM por exemplo, introduz-se a possibilidade de criar elos adicionais sobre uma estrutura de documento previamente definida, permitindo não só o reuso de nós, mas também o reuso de uma estrutura composta de nós e elos (Soares, 2000). Entretanto, apesar das vantagens, em modelos onde os elos estão contidos nas composições, o conjunto de elos que tocam um determinado nó depende da estrutura de composições em que o nó está inserido. Com isso, não consegue-se modelar elos similares aos elos HTML, ou seja, que são válidos independente da estrutura de composições em que um nó esteja inserido. No HTML, qualquer documento, que utilize um nó, inclui sempre o conjunto de elos que está embutido no nó.

Outros modelos não usam composições para representar relações de estruturação, mas sim para representar relações de sincronização. Nesse caso, cada tipo de composição tem uma semântica temporal embutida que será utilizada para apresentar seus componentes. Como exemplo, temos o formato CMIF (van-Rossum, 1993), que mais tarde originou o modelo usado na linguagem SMIL (W3C, 1998c; W3C, 2001d). Essa abordagem torna mais fácil a tarefa de autoria, pois o autor pode especificar com uma única composição o que seria especificado com o uso de vários elos (Rodrigues, 2002a). Por outro lado, a maioria das linguagens oferece um conjunto limitado de composições com semântica de sincronização. No SMIL, por exemplo, há somente três tipos, que são paralela, seqüencial e exclusiva (W3C, 2001d). Como consequência, relacionamentos complexos devem ser construídos através de um aninhamento das composições básicas, o que pode acabar dificultando o processo de autoria. Além disso, o autor é obrigado a estruturar o documento de acordo com sua especificação temporal de apresentação.

A especificação de relacionamentos de sincronização através de elos permite que o uso de composições possa ser dedicado à criação de relações de estruturação entre componentes de um documento. Além disso, o uso de elos para definir o sincronismo temporal dá mais flexibilidade a um modelo hipermídia, pois permite que um autor defina qualquer tipo de relação de sincronização entre um grupo de componentes. Uma composição que contém esses elos tem a sua semântica temporal definida pelos elos, sem precisar ficar limitada a tipos pré-definidos pelo modelo. Por outro lado, o uso de composições com semântica temporal facilita bastante o trabalho do autor por evitar que ele tenha que criar vários elos entre um grupo de nós para obter uma semântica idêntica à da composição.

O cenário ideal seria, portanto, oferecer ao autor todas as possibilidades: o uso de composições para estruturação lógica e o uso de elos para especificar sincronização; e o uso de composições com semântica, por exemplo temporal, fazendo a estrutura lógica casar com a estrutura de apresentação. Melhor ainda se uma linguagem de autoria permitisse a criação de tipos de composições com semântica que pudessem ser utilizados de acordo com a necessidade do autor. Assim, a estrutura de apresentação de um documento poderia se adaptar à sua estrutura lógica, e não o contrário. Esse propósito será seguido nesta tese.

Um modelo hipermídia ideal deveria fornecer o suporte para especificar diferentes tipos de relações entre componentes de um documento, oferecendo bastante expressividade para o autor e ao mesmo tempo facilitando suas tarefas. Um modelo ideal permite o desenvolvimento de uma linguagem de autoria que também satisfaça os mesmos requisitos. Essa é a principal motivação desta tese, que aborda a especificação de relações em modelos hipermídia e em linguagens declarativas, focando-se na expressividade e no reuso para facilitar o trabalho do autor.

Além da importância da expressividade que deve ser oferecida por uma linguagem de autoria, um outro ponto-chave para a utilização de uma linguagem em larga escala é sua simplicidade de uso. Como exemplo marcante desse fato,

tem-se a linguagem HTML, inicialmente padronizada pelo IETF² – *Internet Engineering Task Force* – para especificação de documentos web (Berners-Lee, 1995). O fator crucial do seu sucesso foi o fato de se basear em um modelo hipertexto bastante simples, como já mencionado, de nós e elos de navegação tradicionais. Com a extensa utilização do HTML, as linguagens declarativas voltaram a ganhar bastante importância como ferramentas de autoria de documentos. Um outro fator que contribuiu bastante para o sucesso desse tipo de linguagem foi a padronização da meta-linguagem XML, simplificando o padrão pioneiro SGML (Wilde, 1999) da ISO³ – *International Organization for Standardization* – e facilitando o desenvolvimento de aplicações processadoras de documentos, chamadas de formatadores (*browsers*). A última versão da linguagem HTML (W3C, 2001a), chamada XHTML 1.1, é especificada em XML e apresenta uma estrutura modular, visando facilitar sua evolução e extensão futura.

Nos últimos anos, várias propostas de linguagens declarativas voltadas para a Web e baseadas em XML surgiram. Dentre essas propostas, pode-se destacar três que também se tornaram recomendações do W3C, SMIL (W3C, 2001d), SVG (W3C, 2001e) e XLink (W3C, 2001c). O objetivo principal dessas linguagens é de superar limitações do HTML ou mesmo complementar seus recursos através da introdução de novos módulos, seguindo a proposta de modularização que vem sendo adotada pelo W3C.

A evolução da área de linguagens de autoria hipermídia (*Hypermedia Authoring Language*), chamadas simplificada de **HAL** neste texto, ficou bastante voltada para a implementação da Web nestes últimos anos, pois ela se consagrou como o sistema hipermídia mais difundido e utilizado por milhões de usuários. Enquanto isso, outras áreas da ciência da computação que têm como objetivo comum o desenvolvimento de linguagens de autoria também evoluíram, só que de uma forma mais abstrata, sem se ater à implementação de um sistema específico. Como exemplo, tem-se a área de arquitetura de software (Garlan, 1995b; Shaw, 1996b), classificada como sub-área da engenharia de software, que

² <http://www.ietf.org>

concentrou esforços na definição de linguagens para especificação de arquiteturas de sistemas de software. Linguagens de descrição de arquitetura (*Architecture Description Language – ADL*) são linguagens formais que podem ser usadas para representar a arquitetura de um sistema de software, definindo seus componentes, como eles se comportam e os padrões e mecanismos para interações entre esses componentes (Shaw, 1996b).

Analisando os elementos de linguagem geralmente oferecidos por uma ADL e comparando-os com os oferecidos por uma HAL, percebe-se que existem muitas semelhanças com relação ao tipo de estrutura gerada, quando uma arquitetura de software é especificada ou quando um documento hipermídia é construído (Paula, 1999; Muchaluat-Saade, 2000). Entretanto, pode-se destacar também algumas facilidades presentes em ADLs que não são encontradas no domínio hipermídia, e vice-versa (Muchaluat-Saade, 2001c), mas que se fossem, trariam benefícios para ambos. Dessa forma, outra motivação deste trabalho é introduzir novos conceitos em linguagens de autoria hipermídia de tal forma que algumas facilidades presentes em ADLs também sejam providas para a autoria de hiperdocumentos, visando sempre o aumento do reuso, expressividade e simplicidade de utilização, como discutido nos parágrafos anteriores.

1.1 Objetivos

O modelo NCM – *Nested Context Model* (Casanova, 1991; Soares, 1995; Soares, 2000; Soares, 2000a; Soares, 2003) é um modelo hipermídia com bastante poder de expressão e que possui características suficientes para a criação de uma linguagem que satisfaça os requisitos necessários para a especificação completa de um documento. Uma primeira versão dessa linguagem, chamada NCL, foi proposta por (Antonacci, 2000a; Antonacci, 2000b), satisfazendo todos os requisitos apontados em (Antonacci, 2000c). O NCM oferece nós, elos e nós de composição como entidades básicas do modelo. Nós de composição no NCM agrupam conjuntos de elos e nós, que podem ser composições, recursivamente.

³ <http://www.iso.org>

Comparando essas entidades com as oferecidas por ADLs, nós, elos e nós de composição hipermídia poderiam ser mapeados respectivamente em componentes, conectores e configurações, que representam as entidades básicas fornecidas pela maioria das ADLs. No entanto, analisando com mais detalhes, podemos notar que a analogia não é tão direta e trivial. Existem algumas características interessantes em ADLs que não têm analogias em HALs.

Em muitas ADLs, as relações entre componentes, representadas por conectores, desempenham um papel importante na descrição de uma arquitetura. Conectores são tratados como entidades de primeira classe, ou seja, podem ser nomeados, subtipados e reutilizados (Shaw, 1994; Mehta, 2000), e seu poder de abstração é comparado ao dos componentes.

Um dos objetivos da tese é trazer essas vantagens para o tratamento de relações em linguagens hipermídia com a introdução de uma nova entidade de primeira classe nos modelos e linguagens hipermídia chamada de *conector hipermídia*. O propósito principal de um conector hipermídia, tal como em ADLs, é representar a semântica de uma relação de forma independente dos participantes (nós hipermídia) que irão interagir através da relação.

Conectores hipermídia podem ser utilizados para especificar os diferentes tipos de relação que uma linguagem de autoria deve oferecer. Dentre esses tipos, relações de sincronização são, sem dúvida, as mais complexas. Por esse motivo, um outro objetivo da tese é a especificação de uma linguagem declarativa modular para definição de conectores hipermídia representando relações de sincronização espaço-temporal. Essa linguagem, batizada de XConnector, foi desenvolvida de uma forma genérica, sendo na verdade um módulo que pode ser incorporado em linguagens de autoria hipermídia existentes, como por exemplo, NCL, XLink, XHTML ou SMIL. Como o uso de conectores está relacionado à especificação de elos hipermídia, e este é o principal objetivo da recomendação XLink do W3C (W3C, 2001c), a incorporação de XConnector a XLink recebeu atenção especial, sendo também desenvolvida. Uma das principais vantagens da integração de XConnector à XLink é aumentar o poder de expressão da linguagem XLink, oferecendo a possibilidade de especificação de relações espaço-temporais entre recursos XML através de elos.

Outra facilidade de ADLs, que não é oferecida em HALs, é a possibilidade de especificar estilos arquiteturais. Estilos arquiteturais possuem os mesmos elementos estruturais que uma configuração representando um sistema de software específico, ou seja, componentes, conectores e outras configurações recursivamente. Adicionalmente, acrescentam a possibilidade de definição de restrições arquiteturais ou semânticas entre os elementos que compõem um estilo.

Fazendo uma comparação com linguagens hipermídia, estilos podem definir estruturas de composições que são comuns em vários documentos, representando templates⁴ de nós de composição que poderiam ser reusados. Quando um usuário quiser fazer um outro documento utilizando o mesmo tipo de composição, ele pode utilizar o mesmo template representando a estrutura pré-definida. Exemplos de composições que podem ser especificadas como templates são composições com semântica temporal ou espacial pré-definida. Por exemplo, os elementos *par* e *seq* da linguagem SMIL (W3C, 2001d) podem ser vistos como templates de composição, onde todos os componentes têm relacionamentos conhecidos que especificam sua apresentação em paralelo ou em determinada sequência dada pela ordem de declaração dos componentes. Uma grande utilidade de templates de composição em linguagens hipermídia é o fato do autor poder criar uma composição com a semântica que desejar e reutilizá-la para a criação de seus documentos, sem ficar limitado aos tipos de composição pré-definidos oferecidos por uma linguagem específica. O oferecimento de templates de composição por uma linguagem de autoria permite, dessa forma, atingir a situação ideal de fornecer ao autor várias possibilidades para que ele realize a autoria de relações no documento, comentada na seção anterior.

Para prover a definição de templates de composição, é necessária uma linguagem que forneça suporte à definição de restrições entre elementos de uma composição indicando, por exemplo, tipos de componentes, número máximo de componentes classificados ou não por tipo, tipos de relações entre esses

⁴ A palavra estilo já possui uma semântica associada em hipermídia, que é a de especificação de características de formatação de um documento, seguindo conceitos presentes em linguagens para definição de estilos, tais como DSSSL (ISO, 1996), CSS (W3C, 1998b) e XSL (W3C, 2001f). Por essa razão, usaremos o termo template de composição para representar o conceito de estilo arquitetural em hipermídia.

componentes e como os componentes estão relacionados. Sendo assim, um outro objetivo da tese é a proposta de uma linguagem com esse propósito, batizada de XTemplate, promovendo o reuso e facilitando o processo de autoria declarativa hipermídia. Um requisito importante é que essa linguagem possa ser usada em conjunto com os atuais padrões do W3C, entre eles, XHTML, XLink e SMIL. De forma análoga à proposta de extensão de XLink com XConnector, uma outra proposta de extensão de XLink introduzindo facilidades de XTemplate também foi desenvolvida.

As principais contribuições desta tese são resultantes da introdução dos conceitos de conectores e estilos arquiteturais, presentes em ADLs, em linguagens de autoria hipermídia. Como a área de interesse deste trabalho é sistemas hipermídia e não engenharia de software, a tese se restringirá a identificar as contribuições de HALs para ADLs, não tendo como objetivo desenvolvê-las. Além disso, apesar da representação formal da semântica de documentos e sua validação serem objetivos futuros, esta tese não tem intenção de realizá-las.

Como forma de validação dos conceitos apresentados, todas as propostas de contribuições na área de linguagens de autoria hipermídia apresentadas na tese foram incorporadas à linguagem NCL (Antonacci, 2000a; Antonacci, 2000b), introduzindo as facilidades decorrentes do uso de conectores e templates de composição. Para isso, uma nova versão da linguagem NCL, chamada de NCL 2.0, apresentando uma estrutura modular foi desenvolvida. Como consequência, uma nova versão do modelo conceitual NCM, no qual NCL se baseia, foi especificada para capturar o conceito de conector hipermídia (Soares, 2003). Note que o conceito de template de composição é uma facilidade da linguagem de autoria e não precisou ser introduzido no modelo conceitual em que ela se baseia.

1.2

Organização da Tese

O restante do texto está estruturado da seguinte forma. O Capítulo 2 apresenta os principais conceitos de linguagens de descrição de arquitetura (ADL), comparando-as com linguagens de autoria hipermídia (HAL). Após a comparação, um modelo genérico, que pode ser utilizado tanto para representar estruturas de arquiteturas de software quanto estruturas de documentos

hipermídia, é proposto. Além disso, esse capítulo discute as facilidades de ADLs a serem introduzidas, com vantagens, em HALs e vice-versa.

O Capítulo 3 comenta resumidamente a versão 2.0 da linguagem NCL, apresentando sua nova estrutura modular e discutindo as funcionalidades de cada um de seus módulos. Os módulos de conectores e templates de composição são abordados detalhadamente nos capítulos seguintes. Uma discussão sobre modelagem de documentos na nova versão do modelo NCM também é apresentada nesse capítulo.

O Capítulo 4 aborda a linguagem XConnector para a autoria de conectores hipermídia. Além de apresentar a linguagem, o capítulo propõe uma extensão para o padrão XLink, permitindo o uso de facilidades de XConnector. Finalmente, uma breve discussão de como outras linguagens podem utilizar XConnector para autoria de elos também é incluída.

O Capítulo 5 aborda a linguagem XTemplate para a autoria de templates de composição hipermídia. Além de apresentar a linguagem, o capítulo propõe uma extensão para o padrão XLink, permitindo o uso de algumas facilidades de XTemplate. Finalmente, de forma análoga ao capítulo anterior, uma breve discussão de como outras linguagens podem utilizar XTemplate também é abordada.

O Capítulo 6 descreve a implementação de conectores e templates de composição realizada no sistema hipermídia HyperProp, baseado no modelo NCM. Esse capítulo apresenta, de forma resumida, a estrutura de classes para implementação de conectores no NCM, a implementação dos parsers de conectores e do processador de templates, além de dar uma visão geral do sistema HyperProp implementado em Java.

O Capítulo 7 apresenta as comparações com trabalhos relacionados, realçando as contribuições da tese e apresentando possíveis trabalhos futuros a serem realizados.