

2

Trabalhos Relacionados

Para realizar a especificação das aplicações hipermídia é desejável que estruturas com alto nível de abstração estejam disponíveis para o autor. Para apresentação, por outro lado, é desejável que as estruturas existentes facilitem a preservação dos momentos temporais absolutos para a ocorrência dos eventos definidos nas aplicações. A mesma estrutura utilizada na autoria também poderia ser adotada para o controle da apresentação. No entanto, como a autoria e a apresentação têm diferentes objetivos, elas são normalmente suportadas por estruturas diferentes. Como consequência, todos os sistemas hipermídia têm, pelo menos, duas sintaxes, uma para autoria e outra para a apresentação.

A entrega das aplicações ao sistema de apresentação pode ser realizada através de uma outra estrutura, correspondente a uma sintaxe intermediária, construída com o objetivo de facilitar a transferência das aplicações do ambiente de autoria para o de apresentação. A sintaxe de transferência pode ser utilizada, por exemplo, para que a especificação de uma aplicação seja entregue em partes temporalmente relacionadas para apresentação, permitindo que uma apresentação seja iniciada sem que toda a sua especificação tenha sido recebida. Também é possível, através da sintaxe de transferência, realizar modificações na sintaxe de apresentação de uma aplicação simultaneamente à apresentação, com o objetivo de atualizar, na apresentação corrente, as modificações realizadas pelo autor.

Muitas vezes, a sintaxe de transferência é a própria sintaxe de autoria ou a sintaxe de apresentação. O MPEG-4 Sistemas (ISO/IEC, 2001), por exemplo, define um mesmo formato binário denominado BIFS (*Binary Format for Scenes*) (ISO/IEC, 2001) tanto para a apresentação quanto para o transporte das aplicações (sintaxe de apresentação = sintaxe de transferência). Nos principais STVDI (ABNT, 2009; ARIB, 2005; ATSC 2005; ETSI, 2005), por outro lado, normalmente a sintaxe de transferência é a mesma adotada na autoria (sintaxe de autoria = sintaxe de transferência). Esse é o caso, por exemplo, do ISDB-T_B (ABNT, 2009), onde aplicações NCL (ABNT, 2009; Soares, 2006a) devem ser

recebidas pelos receptores antes do início da apresentação. Além disso, durante a apresentação, comandos, contendo especificações NCL, também podem ser distribuídos aos clientes para modificar a sintaxe de apresentação existente (Costa, 2006).

Considerando que o objetivo desta tese está relacionado ao controle do sincronismo na apresentação, modelos que possam oferecer suporte a esse objetivo são o foco deste capítulo. Assim, modelos voltados à autoria, como, por exemplo, o NCM (*Nested Context Model*) (Soares, 2005a), que serve como base para a especificação em NCL, não são considerados.

Nas próximas seções são analisadas as características de modelos voltados ao controle da apresentação em relação ao controle do tempo, à edição ao vivo e à realização de apresentações distribuídas através de múltiplos dispositivos, que são as principais contribuições deste trabalho. Estruturas utilizadas na sintaxe de transferência também serão abordadas quando tiverem relação direta com as contribuições mencionadas. Características individuais dos modelos não são discutidas em detalhes neste capítulo, porém, o leitor interessado pode encontrar essas informações no Apêndice A.

2.1.

Controle do Tempo da Apresentação e Transmissão

Timeline é um modelo para a definição da sincronização nas aplicações baseado em um eixo do tempo (Bertino, 1998; Blakowski, 1996), onde os pontos de sincronização são associados a momentos temporais absolutos em relação a uma base, por exemplo, o início da apresentação. Esse modelo favorece o controle do tempo, tanto para a apresentação quanto para transmissão, uma vez que todos os momentos temporais das ocorrências dos eventos são conhecidos antes do início da apresentação da aplicação. No entanto, essa forma de sincronização é ineficaz quando aplicada às aplicações hipermídia, uma vez que ela não é capaz de representar eventos com duração variável ou mesmo imprevisível.

Como exemplo da sincronização *timeline*, pode ser mencionado o BIFS, mencionado no início deste capítulo. Para a autoria, o MPEG-4 Sistemas define uma outra linguagem (XMT-O (ISO/IEC, 2001)), onde os relacionamentos entre os eventos são especificados de forma relativa. No entanto, o uso de um modelo

na autoria capaz de representar eventos imprevisíveis não impede a perda da semântica dos relacionamentos entre os eventos, quando traduzidos para o BIFS (padronizado como forma de apresentação e formato de transferência), restando apenas os momentos previstos para a ocorrência dos eventos. Essa conversão se dá pela quebra da aplicação XMT-O em várias cenas BIFS, como se a aplicação fosse dividida em várias pequenas aplicações ligadas.

Na divisão de uma aplicação original, cada uma das aplicações gerada é usualmente formada por um conjunto de mídias associadas a um mesmo *timeline* temporal, completamente previsível. Essas aplicações podem ser iniciadas ou encerradas a partir de eventos externos à aplicação. No MPEG-4 Sistemas, por exemplo, eventos interativos podem ser utilizados para encerrar a apresentação atual e iniciar uma outra apresentação. Essa estratégia também é utilizada nos STVDI europeu (ETSI, 2005), americano (ATSC, 2005) e japonês (ARIB, 2005), onde várias aplicações baseadas em XHTML (W3C, 2002) podem ter suas apresentações encerradas ou iniciadas a partir de eventos interativos. Nesses STVDI, as apresentações também podem ser controladas através de eventos de sincronismo (ISO/IEC, 1998), enviados aos clientes em momentos específicos da exibição do conteúdo audiovisual principal. A solução desses sistemas é, no entanto, pior ainda do que a utilizada no MPEG-4, que pelo menos possui uma linguagem de auto nível (XMT-O) para a especificação da aplicação global (que reúne as várias aplicações do conjunto). No caso desses outros sistemas, cabe ao autor da aplicação realizar a quebra e o controle, ao passo que em MPEG-4 ela pode ser automatizada.

Dividir uma aplicação em várias pequenas aplicações é conveniente apenas para aplicações simples e, mesmo nesse caso, a semântica original da aplicação global pode ser completamente perdida. Em seu lugar tem-se um conjunto de apresentações isoladas executadas no tempo. A semântica da aplicação original permanece apenas na idéia do autor, que, normalmente, passa a ser também o controlador da apresentação, tarefa que, para aplicações com um grande número de mídias e eventos, pode ser difícil. Nos STVDI mencionados, outra possibilidade seria utilizar uma linguagem imperativa para a especificação das aplicações, porém, variáveis, estruturas condicionais e outras características inerentes a esse paradigma de programação dificultam o controle temporal das mesmas.

Na sincronização baseada em eventos, ao contrário da sincronização *timeline*, os relacionamentos em uma aplicação são definidos de forma relativa, sem nenhuma menção ao tempo absoluto de suas ocorrências. Aplicações hipermídia podem ser completamente especificadas e apresentadas através da sincronização baseada em eventos, no entanto, essa forma de sincronização não oferece facilidades para o controle do tempo da apresentação e da transmissão dos dados, uma vez que os momentos de ocorrência dos eventos nas aplicações são desconhecidos.

Como exemplo da sincronização baseada em eventos, pode ser citado o exibidor do sistema HyperProp (Soares, 2000), que realiza a apresentação através de um modelo de execução orientado a objetos. Nesse sistema, os conteúdos das mídias e os seus eventos associados são representados por objetos, que encapsulam suas propriedades e ações (Rodrigues, 2003a). Uma ação executada sobre um objeto pode gerar notificações em outros objetos, que, por sua vez, dão origem à execução de outras ações e, assim, sucessivamente. Esse modelo é suficiente e eficiente para realizar a apresentação das aplicações, porém, não oferece facilidades ao controle temporal, definidas como alvo desta tese no Capítulo 1.

Modelos que utilizam estruturas baseadas em grafos são bastante utilizados para o controle das apresentações (Buchanan, 1992; Chung, 2005; Jourdan, 1998; Little, 1990; vanRossum, 1993; Vuong, 1995). Alguns desses modelos são especializados em relação às suas entidades, seu formato ou em relação à semântica do sincronismo que eles representam. Nesse caso, usualmente, esses modelos possuem classificação própria, como é o caso, por exemplo, do modelo hierárquico (Baecker, 1996; Bulterman, 1991) e das redes de Petri (Chung, 2005; Little, 1990; Vuong, 1995; Woo, 1994; Yang, 2003).

A sincronização hierárquica apresenta limitações tanto para a especificação quanto para o controle da apresentação (Souza, 1997). A sincronização baseada em redes de Petri, por outro lado, possui a formalização necessária para a verificação de muitas propriedades temporais que podem ser utilizadas, tanto no controle do tempo da apresentação quanto da transmissão. Todavia, essa análise formal não é normalmente o foco do controle da apresentação e transmissão das aplicações hipermídia, e sua implementação em tempo de apresentação das aplicações pode ser complexa e resultar em atrasos inaceitáveis.

Diversos outros modelos baseados em grafos, genericamente denominados grafos temporais, podem ser utilizados para controlar as apresentações (Bertino, 1998; Buchanan, 1992; Jourdan, 1998; vanRossum, 1993). O exibidor CMIFed (vanRossum, 1993), por exemplo, utiliza uma estrutura denominada grafo de dependências temporais, construída a partir da estrutura hierárquica de uma aplicação CMIF (CWI *Multimedia Interchange Format*) (Bulterman, 1991), base da especificação da linguagem SMIL (*Synchronized Multimedia Integration Language*) (W3C, 2008b), que é a recomendação W3C³ para a especificação de aplicações multimídia na Web.

No grafo de dependências temporais do CMIFed, os vértices representam o início e o fim dos eventos e as arestas as restrições temporais dos relacionamentos entre dois vértices. Durante a apresentação, os vértices vão sendo marcados. Um vértice especial, que representa o início da apresentação, é o primeiro a ser marcado. Ao ser marcado, as restrições temporais associadas as arestas de saída de um vértice geram atrasos (condições de sincronização temporal) que devem ser individualmente respeitados para cada aresta. Quando todas as arestas de entrada de um vértice têm seus tempos atingidos, esse vértice destino é marcado. A aplicação termina quando um outro vértice, que representa o fim da apresentação, é marcado.

O grafo do CMIFed oferece facilidades temporais como, por exemplo, a sinalização de que o término de uma apresentação foi alcançado (o vértice de fim da apresentação é marcado). No entanto, como todos os vértices folhas do grafo devem ser alcançados, essa estrutura não favorece a representação de aplicações adaptativas, que poderiam levar a apresentações diferentes, dependendo, por exemplo, das escolhas dos usuários ou das suas características, ou ainda das características do sistema de apresentação.

O exibidor Ambulant (Bulterman, 2004; Cesar, 2006) é uma ferramenta mais recente, evolução do CMIFed, proposta como referência para a apresentação de aplicações SMIL. Para controlar a apresentação, o Ambulant constrói um grafo temporal formado a partir dos relacionamentos hierárquicos e dos relacionamentos entre os eventos existentes na aplicação. Diferente do CMIFed, cada vértice do grafo do Ambulant possui uma máquina de estados que controla o estado do

³ World Wide Web Consortium – <http://www.w3.org>

evento de apresentação de um mídia ou de uma composição (Bulterman, 2004). A execução dos eventos é realizada a partir dos relacionamentos hierárquicos ou a partir de transições ocorridas na máquina de estados de um nó. Assim, embora essa estrutura seja suficiente para controlar a apresentação, ela possui características muito próximas da sincronização baseada em relacionamentos entre eventos, identificadas anteriormente no formatador HyperProp e, portanto, não oferece facilidades para o controle temporal das apresentações.

Entre os diversos modelos baseados em grafos disponíveis, o modelo proposto no sistema Firefly (Buchanan, 1992; Buchanan, 1993; Buchanan, 2005) merece destaque em função do controle temporal que ele oferece. Nesse modelo, um conjunto de grafos é utilizado para representar os relacionamentos definidos em uma aplicação. A sequência de eventos previsíveis, a partir do evento de apresentação da mídia que inicia a aplicação, forma um grafo denominado principal. Para cada evento imprevisível, cujo tempo de execução não pode ser calculado antes do início da apresentação, um grafo auxiliar é construído.

Durante a apresentação, cada grafo auxiliar no Firefly é adicionado ao grafo principal tão logo o tempo de execução do evento inicialmente imprevisível que originou esse grafo auxiliar possa ser calculado. Dessa forma, quando o último evento imprevisível da aplicação é conhecido, o grafo temporal é completamente obtido, bem como o *timeline* da aplicação. Essa estrutura é adequada para o controle do tempo da apresentação e, portanto, sua proposta foi o ponto de partida desta tese. No entanto, essa estrutura é incompleta, uma vez que ela não preserva os relacionamentos entre os eventos. O grafo resultante representa apenas os momentos temporais associados à execução dos eventos em uma aplicação e não a semântica dos relacionamentos entre os eventos. Além disso, essa única estrutura não é capaz de representar, simultaneamente, todas as especificações temporais que merecem ser controladas, incluindo, por exemplo, a representação simultânea do controle da apresentação, dos momentos para solicitação dos conteúdos das mídias aos servidores, dos momentos para carregamento dos exibidores etc. O HTG pode ser considerado uma extensão dos grafos Firefly.

2.2. Edição Simultânea à Apresentação

Para atualizar nos clientes as modificações realizadas sobre as especificações das aplicações, uma opção é utilizar a mesma sintaxe de apresentação também para a transferência. Essa é, como mencionada, a estratégia do MPEG-4 Sistemas, onde a sincronização *timeline*, através do BIFS, é utilizada tanto na transferência quanto na apresentação das aplicações. Nesse sistema, durante a apresentação, é possível que os conteúdos das mídias sejam atualizados ou mesmo que objetos sejam inseridos, removidos ou que tenham os valores de suas propriedades modificados. No entanto, quando a sintaxe de transferência é a mesma utilizada na apresentação, as modificações realizadas não são registradas na especificação da aplicação, anteriormente enviada ao sistema de apresentação. Nesse caso, apresentações posteriores poderão não refletir as modificações realizadas. Isso pode acontecer com aplicações XMT-O, quanto a atualização é realizada através do BIFS.

Nos STVDI europeu, americano e japonês, aplicações baseadas em XHTML são transmitidas e apresentadas nos clientes. Durante a apresentação, outras dessas aplicações podem ser transmitidas, tanto para serem exibidas em paralelo quanto para substituir as aplicações existentes. Em ambos os casos, através das modificações nessas aplicações, a semântica da “aplicação original”, construída a partir de várias dessas aplicações XHTML, pode ser modificada simultaneamente à sua apresentação. Eventos de sincronismo também podem ser utilizados na atualização das aplicações. Nesses STVDI, no entanto, para preservar a especificação da aplicação global (que reúne as várias aplicações do conjunto), o autor precisa manter o controle da versão apresentada, que inclui o histórico das aplicações XHTML e dos eventos de sincronismo executados. Essa tarefa, para aplicações com um grande número de mídias, eventos e atualizações, é, no mínimo, ineficiente e muito suscetível a erros. Além disso, esse controle não é transferido ao cliente que, ou perde a semântica global de uma aplicação, ou tem de inferi-la a partir dos comandos da sintaxe de transferência, o que em geral é, no mínimo, muito difícil.

Nos STVDI mencionados, linguagens imperativas também podem ser utilizadas na especificação das aplicações. Nesse caso, a sintaxe de transferência,

a mesma utilizada na autoria, seria formada por especificações de uma linguagem de programação. Linguagens interpretadas, como a linguagem Java⁴ base da especificação imperativa nos STVDI, permitem que modificações sejam realizadas simultaneamente à apresentação. Particularmente, Java possui outras características favoráveis à edição simultânea à apresentação, como a alocação dinâmica de memória, incluindo a remoção de objetos que não são mais necessários (*garbage collector*). No entanto, independente dessas facilidades, pode ser difícil associar as modificações realizadas sobre a especificação de um programa interativo com a sua apresentação. Essa deve ser a razão pela qual os STVDI não permitem modificações simultâneas à apresentação, quando as aplicações são especificadas de forma imperativa.

Nesta tese, uma sintaxe de transferência é proposta com base na especificação utilizada na autoria, a fim de preservar a semântica das aplicações geradas ao vivo. No sistema de apresentação, as modificações recebidas, se relacionadas ao sincronismo temporal, irão provocar alterações na estrutura de apresentação, que corresponde ao HTG da aplicação.

A sintaxe de transferência proposta, que será apresentada no Capítulo 6, é muito próxima à de autoria. Na verdade, essa sintaxe espelha as modificações do autor através de comandos, não sendo adequada, portanto, ao controle da transmissão. Por esse motivo, o HTG também é calculado no ambiente de autoria (servidor), não como uma sintaxe a ser usada na autoria ou na transferência, mas para ser utilizada no controle da transmissão de dados sem solicitação (*pushed data*).

A edição simultânea à apresentação, embora seja necessária à especificação de aplicações hipermídia, como apresentado no Capítulo 1, ainda não é uma funcionalidade facilmente encontrada em ferramentas destinadas à apresentação dessas aplicações (Costa, 2006). No exibidor Ambulant, por exemplo, é necessário que uma aplicação SMIL seja completamente recebida antes do início da sua apresentação. Uma vez iniciada a apresentação nesse exibidor, qualquer modificação na especificação da aplicação não será refletida na sua apresentação.

⁴ <http://www.sun.com/java>

2.3.

Apresentação Distribuída através de Múltiplos Dispositivos

O uso de múltiplos dispositivos em apresentações hipermídia, como mencionado no Capítulo 1, não é uma funcionalidade nova (Cesar, 2006; Cesar, 2007; Cesar, 2008; Dini, 2007; Schroyen, 2008), principalmente quando o controle e a distribuição do conteúdo são realizados de forma centralizada. O exibidor Ambulant, por exemplo, pode atuar como um servidor em uma arquitetura onde é permitido aos dispositivos visualizar uma apresentação comum e selecionar uma parte específica para apresentação individual (Cesar, 2007; Cesar 2008). As ações executadas nos clientes (dispositivos) são transmitidas ao servidor, onde são processadas pelo Ambulant, que pode retornar uma aplicação, contendo a disposição espacial das mídias para apresentação no dispositivo onde a seleção foi realizada, em uma verdadeira arquitetura cliente-servidor (Soares, 2009a).

Quando a distribuição da apresentação aos dispositivos é realizada de forma centralizada, os requisitos para o controle da apresentação são praticamente os mesmos de quando a apresentação é realizada através de um único dispositivo. A principal diferença está na comunicação entre os dispositivos. Os aspectos dessa comunicação (controle da presença dos dispositivos, protocolos de comunicação entre os dispositivos etc.), no entanto, não são o foco desta tese que, embora considere a possibilidade do controle centralizado, está mais relacionada com a realização desse controle de forma distribuída, onde um cliente pode até mesmo tornar-se um servidor, em um verdadeiro controle distribuído hierárquico (Soares, 2009a).

No cenário distribuído hierárquico, faz-se necessário que a estrutura utilizada no controle da apresentação facilite a fragmentação da especificação de uma aplicação durante a sua apresentação. Essas partes, por sua vez, devem ser distribuídas, através de uma sintaxe de transferência, aos dispositivos desejados, cada parte a um dispositivo diferente. Posteriormente, pode ser necessário que essas partes obtidas possam ser novamente fragmentadas e distribuídas e, assim, sucessivamente.

Devido ao fato dos momentos temporais para a ocorrência dos eventos serem previamente conhecidos na sincronização *timeline*, ela pode favorecer a

fragmentação temporal das aplicações. Curiosamente, o MPEG-4 Sistemas não especifica uma forma direta de apresentação através de múltiplos dispositivos, apesar de favorecer a divisão de uma aplicação em várias outras aplicações, bem como a distribuição das aplicações obtidas a partir de uma aplicação original. Para apresentação através de dispositivos móveis, o MPEG-4, na sua parte 20 (ISO/IEC, 2006), sugere a utilização de uma outra linguagem, o LAsER (*Lightweight Application Scene Representation*) que possui construções baseadas, principalmente, no SVG (*Scalable Vector Graphics*) (W3C, 2003) e SMIL.

A sintaxe de apresentação para aplicações LAsER pode variar de acordo com a ferramenta utilizada para a sua apresentação.⁵ É provável, no entanto, considerando as características do SVG e SMIL, que as ferramentas, em geral, utilizem um formato similar ao proposto pelo Ambulant, que não favorece o particionamento temporal das aplicações nem a transferência de partes específicas da aplicação original.

Em relação à apresentação de aplicações SVG, em (Concolato, 2007) os autores sugerem a fragmentação temporal dessas aplicações com o objetivo de otimizar a utilização da memória no sistema de apresentação. No entanto, nesse trabalho são apresentadas apenas aplicações simples e, mesmo nessas aplicações, a estratégia para fragmentação, que eventualmente também poderia ser utilizada na apresentação através de múltiplos dispositivos, não é discutida. Assim, (Concolato, 2007) é citado apenas como um exemplo das muitas possibilidades da fragmentação temporal das aplicações hipermídia, além, evidentemente, da apresentação através de múltiplos dispositivos.

Considerando o suporte limitado que, em geral, é oferecido para que as aplicações possam ser temporalmente fragmentadas durante a apresentação, alguns trabalhos propõem que a própria especificação das aplicações já seja realizada de forma fragmentada. Esse é o caso, por exemplo, do ambiente TERESA (Paterno, 2008), que permite a autoria multimodal de aplicações. Nesse ambiente, as aplicações são especificadas através de uma sintaxe abstrata que pode ser convertida para várias sintaxes de apresentação e de transferência padronizadas para diferentes plataformas (Paterno, 2008). Essa proposta, no

⁵ O MPEG-4 especifica um conjunto de ferramentas que servem como software de referência (ISO/IEC, 2000a) desse padrão, porém, atualmente, não está disponível uma implementação de referência para a apresentação de aplicações LAsER.

entanto, exige que a escolha sobre a distribuição da apresentação seja realizada ainda na autoria e de forma estática, quando seria interessante que essa distribuição pudesse ser realizada de forma dinâmica, durante a apresentação. Realizada de forma dinâmica, a distribuição de uma apresentação pode ser realizada de forma adaptável, quer seja pelas escolhas realizadas pelos usuários ou mesmo em função das características do ambiente, incluindo se existem dispositivos apropriados à apresentação no momento atual.

Nos STVDI que suportam aplicações imperativas, o autor pode especificar que uma aplicação deve ser apresentada através de múltiplos dispositivos. Esse é o caso, por exemplo, do Ginga-J (Souza, 2007), a parte imperativa do *middleware* Ginga, onde classes e métodos baseados em Java (Souza, 2007) estão disponíveis para controlar os dispositivos existentes em um sistema de apresentação. Nesse caso, no entanto, o autor deve implementar todo o controle dos dispositivos e a especificação da fragmentação de uma aplicação, o que pode ser uma tarefa difícil, principalmente se o autor não for um especialista em programação Java com conhecimento da API Ginga-J, no caso, ou da API de outro STVDI específico. Além disso, a especificação do autor para a distribuição das aplicações torna-se parte da aplicação, devendo ser realizada, novamente, para cada aplicação desejada.

Na proposta desta tese, a distribuição e a possibilidade de distribuição são determinadas, declarativamente, no ambiente de autoria. No entanto, a distribuição é hierárquica e ciente de contexto, sendo realizada de forma adaptativa durante a apresentação de uma aplicação, levando em conta inclusive as interações no cliente receptor. Para tanto, a sintaxe de apresentação deve poder ser hierarquicamente particionada e distribuída do receptor ancestral a seus receptores filhos.