

## 5

### Requisitos e Formatos de Documentos Multimídia

A utilização de diversos dispositivos computacionais para a realização de uma apresentação requer, além de uma infra-estrutura de execução que dê suporte a esta atividade, o controle autônomo da apresentação. Muitas vezes em nossos protótipos utilizamos scripts para controlar os diferentes componentes da apresentação; todavia, para que usuários não especializados em programação possam construir suas próprias apresentações, é necessário o uso de uma descrição com um nível de abstração mais alto, como por exemplo um documento multimídia.

A descrição da apresentação, expressa por meio de um documento multimídia, também é útil em um sistema de autoria, que pode utilizar esse documento como forma de representação. Assim, um documento multimídia serve como um elo entre o sistema de autoria e o sistema de execução. Ao sistema de execução damos o nome de formatador [12, 22]. O formatador interpreta e executa um documento multimídia garantindo a sincronização entre as diferentes partes da apresentação conforme especificado no documento.

A seguir são apresentados os requisitos de representação do documento multimídia. Esses requisitos foram identificados através dos cenários apresentados no Capítulo 1 e dos protótipos descritos no Capítulo 3. Posteriormente, apresentamos avaliações dos formatos SMIL [19] e NCL [11, 42]. Finalmente, propomos a utilização do formato NCL com algumas adaptações para dar suporte os cenários propostos de apresentações multimídia distribuídas em Espaços Ativos.

#### 5.1

##### Requisitos

Os requisitos para a representação de um documento multimídia apresentados a seguir foram levantados através dos diversos protótipos construídos (Cap. 3). Nota-se que alguns desses requisitos coincidem com os

observados por Dick C. A. Bulterman sobre as necessidades de representação em um sistema de autoria de apresentações multimídia [57].

**Novos tipos de mídias:** Um documento multimídia deve levar em consideração a possibilidade de que novos tipos de mídias sejam criadas no futuro e precisem ser representadas no documento (e.g. um projetor de hologramas). A inclusão desses novos tipos de mídia requer que uma nova aplicação possa ser incorporada ao sistema e representada no documento. Além disso, essa nova mídia irá conter diferentes estados a serem representados e relacionados com outras aplicações.

**Representação de estados:** Em mídias tradicionais como vídeo e som, é possível unificar a representação dos estados através de um único parâmetro de representação, como o tempo de reprodução desde o início do arquivo. Todavia, em mídias menos tradicionais como um rastreador de posicionamento, são necessárias novas formas de representação de estados que possibilitem armazenar, por exemplo, um conjunto de coordenadas tridimensionais.

**Sincronia entre as mídias:** É necessário que o documento represente a relação de sincronia entre as diferentes mídias. Nota-se que os diferentes formatos de documento multimídia oferecem representações de sincronia bastante distintas. Como veremos, o formato SMIL apresentado na Seção 5.2 oferece como meio de sincronização principal relações pré-definidas e o formato NCL (Seção 5.3) utiliza ligações através de elos.

**Marcos de execução:** Certas condições durante a execução de uma apresentação multimídia, tal como a chegada de uma marca em uma mídia ou a interação direta do usuário sobre a apresentação, fazem parte do plano de execução e precisam ser representadas no documento.

**Representação espacial:** A representação de uma apresentação exige também a especificação de restrições espaciais que indiquem onde cada mídia deverá ser apresentada. Em um ambiente de computação ubíqua isso significa também escolher entre os múltiplos dispositivos disponíveis. Para que um arquivo possa indicar um dispositivo independentemente do ambiente computacional é necessário que o arquivo ofereça alguns parâmetros de busca e identificação. A adaptabilidade do documento para diferentes salas irá depender desses parâmetros.

**Substituição de conteúdo:** A utilização de um dispositivo de visualização (por exemplo projetor, monitor e televisão) pode ser feita por diferentes aplicações em diferentes pontos da apresentação. Por exemplo, ao término de execução de um vídeo, pode-se desejar que uma imagem de fundo seja apresentada. Portanto é necessário permitir a substituição de aplicações (*players*) associadas a um dispositivo em tempo de execução.

**Análise de desempenho:** Uma análise do desempenho do sistema deve garantir que a sincronia e a reprodução do conteúdo de mídia estejam de acordo com sua representação. Uma abordagem é colocar na representação os requisitos de desempenho que cada documento exige (como descreve a versão 2.1 de NCL - Seção 5.3) ou apenas que o sistema deve preparar a apresentação antes de começá-la (e.g. fazendo *buffering*). Em ambientes de computação ubíqua, novos fatores devem ser considerados, como retardos de rede, latência e diferenças de desempenho entre dispositivos.

## 5.2

### O Formato SMIL

A linguagem SMIL [19] (Synchronized Multimedia Integration Language) é um formato para a definição de relações de sincronismo temporal e espacial derivado de XML. Através de eventos e composições, SMIL permite a criação de apresentações multimídia interativas. O autor pode descrever o comportamento de uma apresentação multimídia relacionando *hyperlinks* com objetos de mídia além de descrever a organização espacial da apresentação na tela. Um de seus objetivos é incorporar tais funcionalidades à World Wide Web superando as limitações do HTML.

SMIL define eventos como *begin* e *end* para início e fim da apresentação, *click* para notificação de pressionamento do botão do mouse, entre outros. Além de eventos, SMIL utiliza para sincronia de conteúdo mecanismos de composição como *seq* para apresentação seqüencial, *par* para apresentação paralela e *excl* para exclusiva. Desta forma, SMIL pré-define tipos de composição, o que simplifica a autoria na maioria dos casos, porém limita a relação de sincronização, que deve utilizar os tipos de composição pré-definidos. Para a composição de documentos mais complexos, a representação requer um alto nível de aninhamento dos tipos básicos. X-SMIL [58] visa a solução dessas questões e de outras baseando-se na linguagem NCL.

### 5.3

#### O Formato NCL

NCL [11, 42] (Nested Context Language) é uma linguagem declarativa modular baseada em XML que, como SMIL, possibilita a representação de documentos multimídia. Em NCL, os objetos de mídia são considerados nós de uma apresentação. Além desse tipo de nó, nós de composições em NCL, assim como em SMIL, podem conter um conjunto de nós, que podem ser objetos de mídia ou outros nós de composição.

Todavia, em NCL, as composições não possuem semântica temporal mas de inclusão. A semântica temporal é dada através de seus elos. Por exemplo, para compor em NCL a relação de seqüência (*seq*) definida por SMIL, basta que elos de ligação sejam colocados de forma a iniciar o nó seguinte da seqüência após o término da execução de um nó.

A ligação dos elos aos nós requer a utilização de pontos de interface do tipo âncora. Âncoras são fragmentos do objeto de mídia. Por exemplo, uma âncora no nó texto pode ser uma palavra. Uma âncora numa figura seria uma determinada região da figura. Uma âncora no nó de áudio seria um intervalo de tempo da música (e.g. de três a cinco segundos). Além de âncoras, os objetos de mídia também podem possuir atributos. Atributos são valores referenciados através de nomes que podem ser consultados e alterados em um objeto de mídia.

As ligações de nós de composição, por outro lado, são feitas através de um tipo especial chamado porta, que define pontos de entrada e de saída tratando a composição como se fosse um objeto único composto de seus nós internos. Assim, a ligação com uma composição pode ser feita através de um elemento do tipo porta ou através de uma âncora. Para permitir a conexão entre nós dentro de uma composição a nós fora dela, um tipo especial de arco é necessário, a que chamamos mapeamento (*map*). Note que os mapeamentos não representam interações entre componentes ou entre conectores (Seção 5.3.1), mas servem apenas para exportar pontos de interface internos para a interface externa de uma composição.

A vantagem de oferecer a criação de mapeamentos e não permitir uma conexão direta entre elementos de composições distintos é satisfazer à propriedade de composicionalidade, facilitando mecanismos de verificação de consistência das estruturas criadas [34].

A Figura 5.1 representa duas composições COMP1 e COMP2. COMP1 está interligada a COMP2 de duas formas: através do ponto i e do ponto j. O ponto i demonstra uma porta de mapeamento ligando i a k. O ponto j

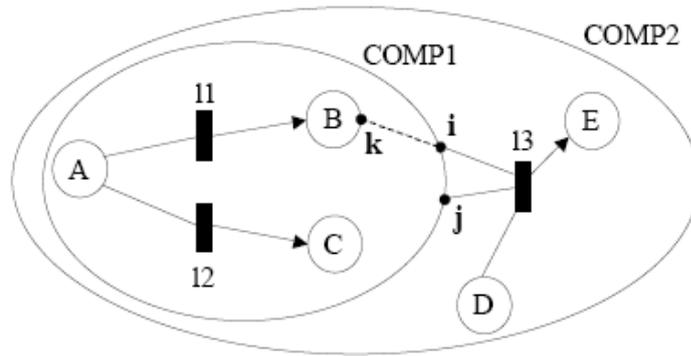


Figura 5.1: Nós de composição com ligações do tipo mapeamento e porta representa apenas um elo do tipo porta.

### 5.3.1 XConnector

O conceito de conector hipermídia (XConnector) desenvolvido por Débora Christina Muchaluat Saade [23, 35] cria recursos para que o usuário possa descrever e estender elos. Um XConnector descreve uma relação de sincronização independente da descrição dos nós participantes e das relações entre eles. Esta separação permite que conectores hipermídia sejam aproveitados múltiplas vezes e com diferentes participantes.

As relações de sincronização previstas pelo XConnector se dividem em relações causais e de restrição. Em uma relação causal, uma condição deve ser satisfeita para que uma ação seja executada. Um exemplo deste tipo de relação é quando a reprodução de um vídeo chega a um determinado ponto, o que pode causar uma ação em outra aplicação, como a troca de página em um navegador web. As relações de restrição, que não possuem nenhuma causalidade envolvida, também podem ser representadas pelo XConnector. Como exemplo desse tipo de relação, um documento pode especificar que um nó da apresentação deve terminar ao mesmo tempo em que outro começa.

A definição de um conector é feita por um conjunto de papéis e uma cola (*glue*). Os papéis determinam a função dos participantes da relação, por meio de eventos e de transições do estado de execução (preparado, ocorrendo e pausado [35, 42]). Os tipos básicos de eventos em XConnector são: *presentation*, que apresenta um conjunto de unidades de informação de um objeto de mídia; *mouseClick*, que informa o clique do mouse sobre um conjunto de unidades de informação de um objeto de mídia; *mouseOver*, que especifica o posicionamento do mouse sobre um objeto de mídia; *focus*,



### 5.3.2 Modularização da Linguagem NCL

Diversos módulos são definidos pela linguagem NCL. A tese de doutorado de Débora Christina Muchaluat Saade [42] apresenta uma lista completa dos módulos de NCL 2.0. A seguir apresentamos exclusivamente os módulos de relevância para este trabalho.

A área Components contém dois módulos chamados *BasicMedia Module* (que é idêntico ao *BasicMedia Module* de SMIL 2.0) e *BasicComposite Module*. O *BasicMedia Module* define os elementos de mídia *animation*, *audio*, *img*, *text*, *textstream*, *video* e *ref*. O módulo *BasicComposite Module* é responsável pela definição de nós de composição em documentos NCL, que já foram explicados na Seção 5.3.

A área Interfaces permite a definição de pontos de interface de nós que serão usados para a criação de elos. Essa área está particionada em quatro módulos, um chamado *MediaInterface*, que permite a definição de âncoras para objetos de mídia, outro chamado *CompositeInterface*, que permite a definição de âncoras e portas para nós de composição, um terceiro chamado *AttributeInterface*, que permite a definição de atributos de nós como pontos de interface, e um último módulo chamado *SwitchInterface*, que não é relevante para este trabalho.

O módulo *MediaInterface* define o elemento *area*. Esse elemento permite a definição de âncoras que representam porções espaciais, temporais e textuais, entre outros. O segundo módulo da área Interfaces é o módulo *CompositeInterface*, que define um elemento chamado *areaComposite*, o qual pode ser usado para a criação de âncoras em nós de composição. O elemento *areaComposite* tem um atributo chamado *componentList*, que contém uma lista de identificadores de nós componentes da composição. Além do elemento *areaComposite*, o módulo *CompositeInterface* também especifica o elemento porta, que define uma porta da composição com seu respectivo mapeamento para um ponto de interface de um de seus componentes. O terceiro módulo da área Interfaces é o *AttributeInterface*, que define um elemento chamado *attribute*, o qual pode ser usado para definir um atributo de um nó como um de seus pontos de interface. O elemento *attribute* tem um atributo chamado *name*, que indica o nome do atributo em questão.

A área Layout define um módulo apenas, o *BasicLayout*, que define como um layout de uma apresentação pode ser construído através de regiões da tela de um computador. Um modelo de posicionamento abstrato é empregado. Maiores detalhes são apresentados por Marcel Stanley Albuquerque

de Moura [24].