

Validação da Estratégia de Adaptação Dinâmica de QoS do Projeto ServiMídia

Luiz Fernando Rust C. Carmo
Luci Pirmez

{rust, luci}@nce.ufrj.br

Roberta Lima Gomes
Renata Falcão Correa
Reinaldo de B. Correia

beta@laas.fr
{refalcao,Reinaldo}@posgrad.nce.ufrj.br

Núcleo de Computação Eletrônica

Universidade Federal do Rio de Janeiro

ABSTRACT

Distributed multimedia systems are applicable in various domains such as Distance Learning. Since multimedia documents may comprise continuous medias, the presentation of those documents may require vast system resources due to huge amount of data and to their QoS retrains. The availability of resources in a distributed environment at presentation time is variable and potentially unpredictable. Hence, it can happen that there are not sufficient resources to render a multimedia document according to its specification, resulting in an arbitrary decrease of presentation's quality. To handle resource scarcity, multimedia documents can be made adaptable to different resource availability. This article presents the results and assessment in terms of latency introduced by an adaptation strategy and a semantic verification tool for adaptive multimedia document (SCL). The adaptation strategy and the mechanisms used in the tool implementation had been conceived inside the ServiMidia project [1].

RESUMO

Os sistemas multimídia distribuídos empregam uma tecnologia aplicável em vários domínios como, em particular, os ambientes de ensino a distância. Considerando que os documentos multimídia são normalmente compostos de mídias contínuas, a apresentação destes documentos pode demandar uma grande quantidade de recursos devido às dimensões dos dados e às restrições de QoS das mídias. A disponibilidade de recursos em ambientes distribuídos no instante da apresentação é variável e imprevisível. Desta forma, os recursos disponíveis para a apresentação do documento multimídia podem ser insuficientes, resultando em uma degradação arbitrária da qualidade da apresentação. Uma forma de se tratar a escassez de recursos é tornar os documentos multimídia adaptativos. Este trabalho apresenta os resultados e avaliação da latência de uma estratégia de adaptação e uma ferramenta de verificação semântica de documentos multimídia adaptativos (SCL). Tanto a estratégia de adaptação quanto os mecanismos utilizados na implementação desta ferramenta foram concebidos dentro do projeto ServiMidia[1].

1. INTRODUÇÃO

De uma forma geral, as aplicações multimídia distribuídas apresentam três grandes dificuldades [2]: (i) se comparadas às aplicações textuais tradicionais, as aplicações multimídia exigem muito mais banda de transmissão; (ii) aplicações multimídia apresentam tráfegos com característica de tempo real, onde os dados (e.g. áudio e vídeo) devem ser apresentados continuamente na mesma taxa em que foram produzidos, limitando o atraso de transmissão, bem como a variação deste atraso; e (iii) os fluxos de dados multimídia, normalmente, apresentam rajadas, o que pode causar a perda de dados devido à falta de espaço em memória (*buffers*) no sistema local ou nos pontos intermediários da rede.

Uma vez que as atuais redes de computadores (e.g., Internet) utilizam-se do paradigma do melhor esforço (*best effort*), escassez de recursos pode ocorrer durante a apresentação de documentos multimídia distribuídos ocasionando a danificação ou até mesmo a interrupção da apresentação.

Objetivando resolver esta situação conflitante, o projeto ServiMídia[1] adota um esquema no qual torna as aplicações multimídia adaptativas. Esse esquema de adaptação que é realizado na máquina cliente utiliza um arquivo de configuração SMIL para coordenar a apresentação multimídia e, também, um arquivo de adaptação SCL para coordenar a adaptação da mesma apresentação quando da ocorrência da falta de recursos na rede. Entretanto, a linguagem SCL sendo muito flexível permite que ocorram inconsistências na especificação do comportamento adaptativo da apresentação. Logo, faz-se necessário que o arquivo SCL seja previamente verificado por uma ferramenta (SCL Semantics) para garantir que o processo de adaptação seja realizado corretamente. Outro ponto importante é verificar se o atraso introduzido nesse processo de adaptação torna esse esquema inviável.

O artigo está estruturado em 6 seções. A seção 2 apresenta além dos conceitos, uma descrição sumária do projeto ServiMídia. A seção 3 descreve a metodologia utilizada e os resultados obtidos na medição da latência do processo de adaptação adotado. A seção 4 enumera os tipos de inconsistências mais comumente encontradas em documentos SCL. A seção 5 descreve a ferramenta de verificação de documentos adaptativos (SCL Semantics) abordando seus requisitos, implementação e interface. Finalmente, a seção 6 encerra o artigo apontando algumas conclusões.

2. CONCEITOS

Todos os mecanismos, estratégias e conceitos em que o presente trabalho se fundamenta foram desenvolvidos no projeto ServiMídia[1].

2.1 - Projeto ServiMídia

O projeto ServiMídia [1] é um projeto de pesquisa na área de redes de computadores e de ensino à distância. A autoria, o armazenamento e a recuperação de documentos multimídia fazem parte do escopo deste projeto. O objetivo básico está na necessidade de se acomodar em um mesmo ambiente multimídia de ensino (baseado no paradigma cliente/servidor) usuários com diferentes perfis de sistema e recursos de comunicação.

Esse projeto tem como proposta tornar as aplicações multimídia adaptativas, de forma a contornar, dinamicamente, o problema da escassez de recursos. Dentro dessa proposta, é definida uma estratégia de autoria e apresentação de documentos adaptativos baseada na especificação de arquivos de controle associados a documentos multimídia convencionais. Um arquivo de controle contém, portanto, as informações necessárias para a realização da adaptação de um documento multimídia. Para a especificação desse arquivo, foi criado o SCL (*SMIL Control Language*), uma linguagem descritiva baseada em XML. Os elementos do

SCL permitem a especificação de parâmetros de QoS, assim como a definição de mídias alternativas e das condições para que as mesmas sejam ativadas durante uma adaptação.

2.2 - Dependências Condicionais – Objeto Link

Para realizar uma adaptação no nível da aplicação, realizando uma reestruturação do documento como um todo, este trabalho baseia-se no conceito de dependências condicionais entre os diferentes objetos.

Para a especificação de dependências condicionais[3], é utilizado o conceito de objeto *link* (*link* condicional). Um *link* consiste de um *LinkSource* (origem do *link*) e um *LinkTarget* (“alvo” do *link*). O *LinkSource*, por sua vez, consiste de uma lista de *SourceConditions* (condições associadas a diferentes objetos de mídia), as quais são combinadas através de uma expressão lógica definida no *SourceDexpr* (definindo a condição de disparo ou ativação do *link*). O *LinkTarget* é formado por uma *TargetAction*, a qual é associada a apenas um objeto e é executada quando o *link* é disparado. Dentro da linguagem SCL são definidos dois tipos de *links*: *startlink* e *stoplink*, para apresentar e interromper um objeto, respectivamente.

2.3- Documentos SMIL e SCL

Além dos módulos da arquitetura, os outros dois componentes vitais são os documentos SMIL (*Synchronized Multimedia Integration Language*)[4] e SCL (*SMIL Control Language*). No documento SMIL, a coordenação da apresentação multimídia está especificada. No documento SCL, estão definidos todos os requisitos para a coordenação da adaptação da apresentação multimídia.

A linguagem SMIL, por ser baseada em XML, é aberta e flexível. E da forma como foi definido o seu DTD (*Document Type Definition*), sua sintaxe ficou simples e intuitiva. Esses foram os principais motivos que levaram a comunidade WWW a adotá-la para a especificação de documentos multimídia de forma distribuída. Para a especificação de documentos multimídia dentro do ambiente ServiMídia foi adotado, portanto, o SMIL 1.0 (o SMIL 2.0 não foi escolhido por estar ainda em desenvolvimento).

SCL é uma linguagem de marcação baseada em XML que permite a especificação de dados estruturados para descrever as informações de controle de um documento adaptativo. Para a definição do SCL[5] foi criado um DTD onde são especificados todos os elementos da linguagem (*tags*, ou marcas) e as regras de utilização destes elementos.

2.5 - Arquitetura do Sistema

Para a verificação da proposta apresentada neste trabalho, foi desenvolvida a arquitetura[5, 6] de um sistema para apresentação distribuída de documentos adaptativos baseada no paradigma cliente-servidor. Os elementos dessa arquitetura são: (i) os servidores de mídia (*Media Servers*), responsáveis pelo armazenamento das mídias e dos documentos que fazem referência às mesmas; e (ii) os clientes (*Media Client*), responsáveis pela apresentação do documento multimídia, podendo apresentar diferentes tipos de características de sistema (terminal, acesso à rede, etc.). Nesta arquitetura, todo o controle é realizado nas pontas do sistema sendo os elementos de rede intermediários desconsiderados.

O *SPlayer* apresenta três funções principais: (i) solicitar um documento de apresentação SMIL; (ii) solicitar aos servidores (via RTSP [7]) a transmissão (via RTP [8]) das mídias referenciadas pelo documento SMIL; (iii) passar ao QoS Manager informações de monitoramento da apresentação.

O módulo QoS Manager além de monitorar os fluxos das sessões RTP/RTCP para controlar o chaveamento entre as variantes de QoS para um mesmo fluxo (adaptação suave), verifica a existência de algum arquivo SCL associado ao arquivo SMIL que será apresentado para em seguida enviá-lo ao Adapter.

O módulo Adapter apresenta duas funcionalidades importantes dentro da arquitetura do sistema, descritas a seguir: (i) Receber e interpretar o arquivo de controle associado ao respectivo documento multimídia, no instante inicial de uma apresentação. As informações de QoS obtidas deste arquivo são repassadas ao *QoS Agent*, que irá utilizá-las para o monitoramento dos fluxos de mídias estabelecidos pelo *SPlayer*. (ii) Reestruturar, em função do arquivo SCL associado e quando sinalizado pelo *QoS Agent*, um novo documento SMIL efetuando o segundo mecanismo de adaptação (adaptação forte).

3. AVALIAÇÃO DA ESTRATÉGIA DE ADAPTAÇÃO

O tempo de resposta do sistema durante o processamento de adaptação em uma apresentação multimídia é uma característica importante para a verificação da viabilidade da estratégia de adaptação. Dentro do Projeto ServiMídia, o atraso introduzido pela rede não é abordado. Entretanto, como parte do trabalho apresentado em [9], foram realizados testes em um protótipo cliente-servidor para observar o comportamento dos mecanismos de adaptação no nível de chaveamento de fluxos. Os resultados destes testes mostraram que o chaveamento de fluxos (executado tanto durante a adaptação suave quanto durante a adaptação forte) introduzia um atraso bastante aceitável. No entanto, o chaveamento de fluxos ocorre somente após a substituição da apresentação original pela apresentação adaptada, representando apenas uma das etapas da adaptação forte. Portanto, para contabilizar a latência total introduzida pelo mecanismo de adaptação forte, também deve ser considerado o atraso introduzido durante o processamento e a geração do documento adaptado.

Nesta seção, são descritos os resultados experimentais e análise referentes ao atraso gerado no nível de documento, realizada através da avaliação do funcionamento do Adapter que é o módulo responsável pela criação em tempo de execução de documentos multimídia. O leitor interessado pode consultar [9] para verificar os resultados referentes ao atraso gerado no nível de chaveamento de fluxos introduzido pelos mecanismos de adaptação quando o chaveamento entre streams de mídia é realizado para executar a adaptação suave (chaveamento entre variantes de QoS de uma mesma mídia) e a adaptação forte (chaveamento entre diferentes tipos de mídia resultante da substituição da apresentação original pela apresentação adaptada).

Considerando que o Adapter deve (i) processar os arquivos SMIL e SCL (para extrair as informações de adaptação) (ii) realizar uma busca nas informações de controle, (iii) reestruturar a apresentação (iv) e construir um novo documento multimídia em tempo de execução, o atraso introduzido por este módulo pode ser bastante crítico. Para otimizar o processo de adaptação forte, o Adapter faz a interpretação (*parsing*) dos arquivos SMIL e SCL no início da apresentação, mantendo em memória as estruturas de dados referentes a estes arquivos até que alguma falha de QoS seja sinalizada. Apesar de haver uma utilização maior de memória, ganha-se em processamento durante a execução da adaptação.

A partir do momento em que é sinalizado, o Adapter realiza a devida busca nas estruturas em memória para descobrir quais mídias devem ser interrompidas e quais alternativas devem ser iniciadas. Em seguida, ele altera a estrutura da apresentação original, criando a apresentação adaptada. Para verificar o tempo de resposta durante a execução desse processo, foram realizadas medidas de latência no módulo Adapter.

O Adapter foi implementado em Java, e os testes foram rodados em uma estação Pentium-II 250MHz com 128MB de memória e sistema operacional Windows NT

Workstation 4.0. Basicamente, durante os testes foram executadas chamadas simulando falhas de QoS durante a apresentação de aplicações testes. Com isso, foram medidos os intervalos entre cada chamada e o instante em que o Adapter concluía a criação de cada documento adaptado.

O atraso do processo de adaptação realizado pelo Adapter está diretamente relacionado ao tamanho das estruturas de dados, obtidas a partir dos arquivos SMIL e SCL: inicialmente, é realizada uma busca pelos links condicionais a serem disparados (quanto maior o número de links definidos no documento SCL, maior o número de passos realizados durante a busca); com base no resultado dessa busca, a estrutura do documento original é alterada (o custo computacional depende do número de elementos especificados no documento SMIL original).

É intuitivo afirmar que a medida que o número de links condicionais aumenta, o documento adaptativo torna-se mais complexo. Portanto, para medir latência do processo de adaptação executado pelo Adapter, foi criado um conjunto de documentos adaptativos (arquivos SMIL e SCL). Para cada documento adaptativo foi definido um número diferente de links condicionais, entre 1 e 28 links, representando as dependências condicionais entre os objetos de mídia. A apresentação de documentos com número de links variados permite, portanto, a observação da latência do processo de adaptação em função da complexidade do documento adaptativo.

Após a autoria de vários documentos adaptativos, verificou-se que, quanto maior o documento multimídia, maior o número de links condicionais especificados. Desta forma, os documentos que possuem mais de 20 links condicionais tornam-se muito grandes e complexos. A criação de documentos com essa dimensão pode torna-se uma tarefa complicada para o autor, mesmo que seja utilizada alguma ferramenta de autoria. Nesse caso, apesar de não ser uma limitação da estratégia de adaptação, é aconselhável que o autor subdivida a apresentação em documentos que contenham até 20 links condicionais, para que o mesmo possa ter um maior controle e organização da apresentação. O limite superior no número de links condicionais para os testes foi adotado, então, com base nesse fato.

Durante a apresentação de cada documento adaptativo, foi simulada a falha de QoS de um de seus objetos de mídia com o objetivo de disparar o processo de adaptação forte. Para simular situações de pior caso, o objeto de mídia escolhido para sofrer a falha de QoS seria o que disparasse o maior número de links condicionais. Uma vez que o algoritmo de busca do Adapter é reiniciado sempre que um novo link condicional é disparado, a quantidade de links disparados durante a busca degrada seu desempenho.

Para realizar as medidas desejadas, é necessário medir o intervalo de tempo entre o evento de disparo da falha de QoS (evento "QoS failure") e o evento representando o início da apresentação adaptada (evento "Replay-Execution"). Este último evento é sinalizado pelo Adapter quando o mesmo finaliza a criação do documento adaptado e envia uma mensagem ao SPlayer submetendo o novo arquivo SMIL. No entanto, durante o início dos testes, verificou-se que a duração da adaptação para um mesmo documento adaptativo variava bastante. Na verdade, isso ocorria principalmente devido ao ambiente de implementação e execução dos testes. A JVM (Java Virtual Machine) utilizada para executar os testes implementa um mecanismo de otimização baseado em cache para os códigos das classes: após as rodadas iniciais de um mesmo teste, a JVM passa a manter em memória o código das classes (e até de objetos) utilizados na aplicação. Conseqüentemente, as últimas execuções de um teste apresentavam um tempo de resposta menor do que as primeiras.

Para a obtenção de um valor médio confiável, foram medidos os atrasos durante diferentes execuções do processo de adaptação de três dos documentos criados (possuindo 2, 10 e 20 links condicionais). Para cada rodada i ($i=[1..20]$) foi obtida a medida do atraso A_i .

Para cada conjunto $c_i = [A_1..A_i]$ foram calculados a média \bar{A}_i e o desvio padrão σ_i . Na Tabela 1 são mostrados os valores de A_i , \bar{A}_i e σ_i obtidos. Pode-se notar que para $i \geq 15$ os valores de \bar{A}_i e σ_i se estabilizam. Com base nesses resultados, as medidas de latência do processo de adaptação foram repetidas 15 vezes para cada documento adaptativo.

Nº DE RODADAS (i)	2 links condicionais			10 links condicionais			20 links condicionais		
	Atraso (A_i) (ms)	Média (\bar{A}_i)	DP (σ_i)	Atraso (A_i) (ms)	Média (\bar{A}_i)	DP (σ_i)	Atraso (A_i) (ms)	Média (\bar{A}_i)	DP (σ_i)
1	70	70.000	---	110	110.000	---	161	161.000	---
2	50	60.000	14.142	120	115.000	7.071	170	165.500	6.364
3	61	60.333	10.017	130	120.000	10.000	171	167.333	5.508
4	60	60.250	8.180	131	122.750	9.845	200	175.500	16.941
5	120	72.200	27.644	130	124.200	9.121	201	180.600	18.582
6	70	71.833	24.742	100	120.167	12.813	201	184.000	18.590
7	40	67.286	25.591	60	111.571	25.572	210	187.714	19.610
8	60	66.375	23.832	71	106.500	27.682	211	190.625	19.935
9	20	61.222	27.128	90	104.667	26.472	230	195.000	22.804
10	70	62.100	25.727	90	103.200	25.385	220	197.500	22.907
11	60	61.909	24.415	60	99.273	27.379	231	200.545	23.964
12	60	61.750	23.285	70	96.833	27.439	240	203.833	25.530
13	40	60.077	23.096	100	97.077	26.285	241	206.692	26.528
14	40	58.643	22.829	80	95.857	25.663	270	211.214	30.592
15	50	58.067	22.112	60	93.467	26.406	300	217.133	37.344
16	80	59.438	22.054	100	93.875	25.563	231	218.000	36.085
17	40	58.294	21.868	90	93.647	24.769	190	216.353	35.561
18	70	58.944	21.394	120	95.111	24.819	200	215.444	34.664
19	30	57.421	21.241	80	94.316	25.429	230	216.211	34.552
20	80	58.550	21.254	90	94.100	24.797	220	216.400	35.320

Tabela 1 - Atrasos medidos durante a adaptação de três documentos com números distintos de links condicionais da tese

A Figura 1 mostra os valores médios e os desvios padrões do atraso introduzido pelo processo de adaptação de documentos possuindo diferentes números de links condicionais. Como era previsto, o gráfico mostra que a complexidade do documento exerce uma influência considerável no desempenho do processo.

Por consequência da grande variação dos valores medidos durante os testes, os desvios padrões calculados não são ideais, em um pior caso chegando a atingir 35% do valor da respectiva média calculada. Apesar dessa grande variação, todos os atrasos medidos foram inferiores a 400 ms. Se for considerado que existe uma expectativa de que o autor defina em torno de 20 links condicionais por documento, o desempenho constatado é ainda melhor. Neste caso os atrasos introduzidos permanecem inferiores a 250 ms. Estes resultados permitem a conclusão de que a geração, em tempo de execução, de novos documentos multimídia adaptados é viável.

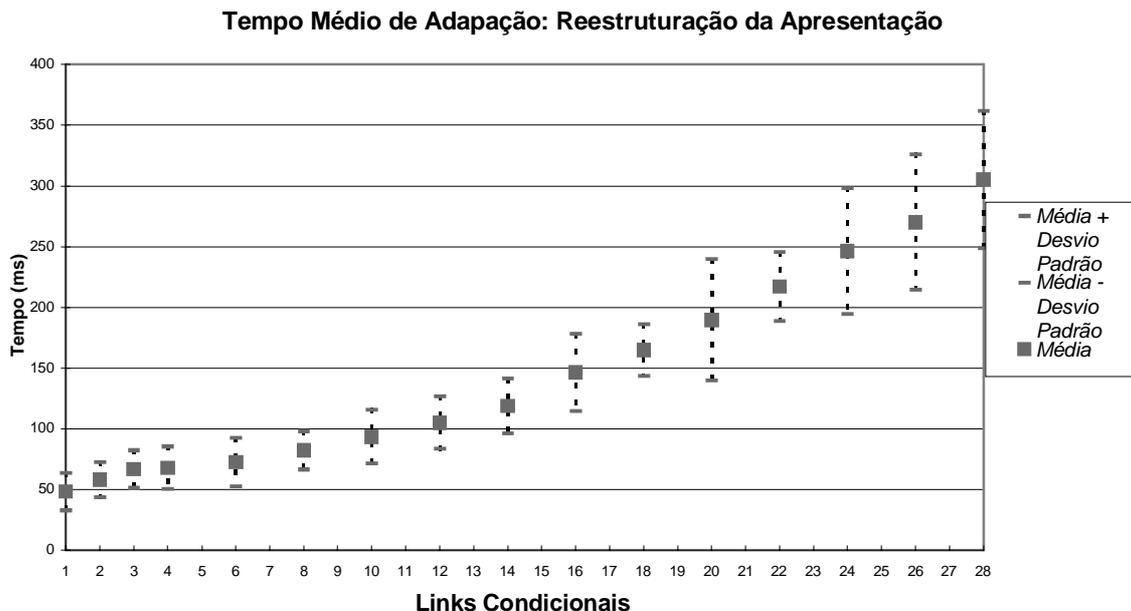
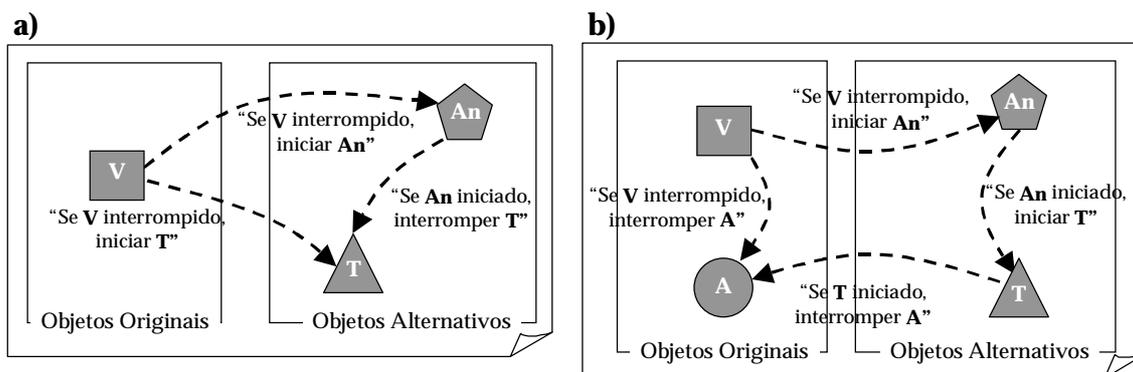


Figura 1 – Atraso introduzido na adaptação forte pelo número de links condicionais

4. CONSISTÊNCIA DAS INFORMAÇÕES DE ADAPTAÇÃO

Após a especificação do SCL, verificou-se que a flexibilidade apresentada pela linguagem permite ao autor especificar informações inconsistentes no arquivo de controle. O esquema adotado de definição de links condicionais entre pares (ou grupos) de mídias permite que ocorra o encadeamento desses links, o que, em alguns casos, pode representar algum tipo de inconsistência semântica nas informações de adaptação. Essas inconsistências podem causar um comportamento indesejável dos mecanismos de adaptação (mais especificamente, do mecanismo de adaptação forte). As seguintes inconsistências podem ser apontadas e detectadas através de uma ferramenta de verificação.

(i) O encadeamento dos links condicionais definidos entre as mídias originais e as mídias alternativas poderá gerar caminhos de links distintos chegando a uma mesma mídia, como ilustrado na Figura 2. Isto pode apresentar duas conseqüências: a ativação e desativação de



uma mesma mídia alternativa (Figura 2.a) ou interrupção/ativação de uma mídia já interrompida/iniciada (Figura 2.b).

Figura 2 - Caminhos de links distintos chegando a uma mesma mídia

(ii) A expressão lógica de um link condicional a qual condiciona ativação do mesmo pode nunca se tornar verdadeira. Neste caso, o link condicional não será disparado em nenhuma

situação, representando, por exemplo, a especificação de mídias alternativas jamais ativadas (Figura 3).

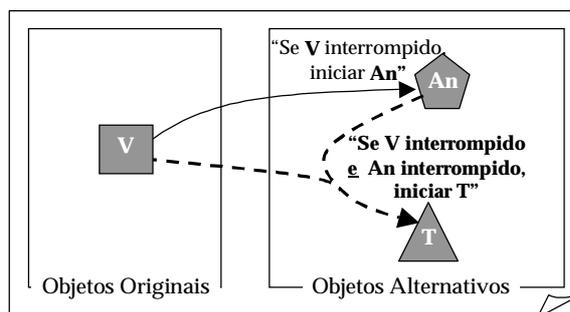


Figura 3 - Link condicional nunca disparado

(iii) A definição de links condicionais entre mídias apresentadas em diferentes instantes da apresentação, ou iniciadas em conjunto mas possuindo durações distintas, pode causar a interrupção de mídias já apresentadas. A Figura 4.a mostra o timeline da apresentação de um documento multimídia. Na Figura 4.b são ilustrados os links condicionais para o respectivo documento. De acordo com as informações de adaptação, se ocorrer uma falha de QoS em “A”, ela será substituída por “C”, assim como a mídia “B” será interrompida e substituída por “D”. Seja t_f o instante da apresentação em que ocorre uma falha de QoS na mídia “A” e sejam t_A e t_B os instantes de finalização das mídias “A” e “B”, respectivamente. Para $0 \leq t_f \leq t_B$, o processo de adaptação é executado normalmente. No entanto, quando $t_B < t_f \leq t_A$, a mídia “B” não está mais ativa. Portanto, o processo de adaptação interrompe uma mídia já apresentada. Se o atributo “restart” definido para a mídia “A” possuir os valores “document” ou “media”, a adaptação ocorre da forma desejada, uma vez que a mídia “D” (alternativa de “B”) será apresentada ao usuário. No entanto, para “restart=failure”, o processo de adaptação apresenta uma inconsistência, uma vez que ele interrompe uma mídia já apresentada (“B”) e ativa uma mídia alternativa que não será exibida (“D”) durante a apresentação do documento adaptado.

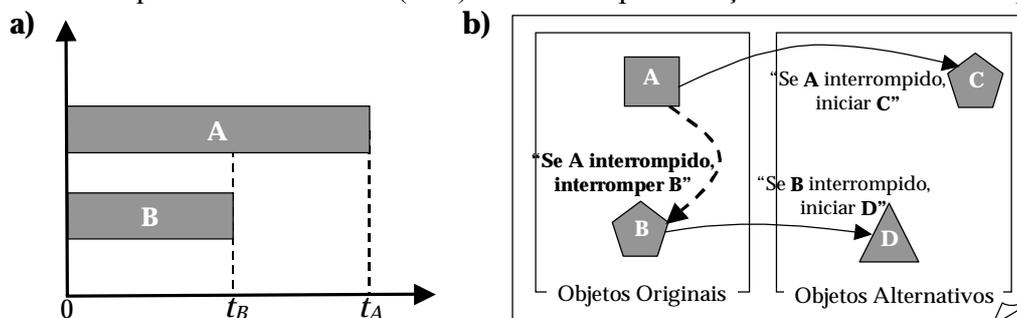


Figura 4 - Link condicional desativando mídia já apresentada, para $t_B < t_f \leq t_A$

5. FERRAMENTA DE VERIFICAÇÃO SEMÂNTICA (SCL SEMANTICS)

Como foi apresentado na sessão anterior, documentos multimídia adaptativos podem apresentar inconsistências referentes às informações de adaptação. Para auxiliar o autor durante a criação dos documentos, mais especificamente, durante a especificação das dependências condicionais, foi desenvolvida uma ferramenta gráfica (“SCL Semantics”) que permite a visualização dessas inconsistências semânticas. Após a criação dos arquivos SMIL e SCL, o autor executa a ferramenta que, com base na simulação de falhas de QoS, executa todas as adaptações possíveis do documento e aponta as inconsistências identificadas.

5.1 - Análise de Requisitos

O desenvolvimento da ferramenta “SCL Semantics” teve como principal objetivo automatizar a identificação de inconsistências semânticas definidas no documento adaptativo. Durante a análise de requisitos foram levados em consideração os três tipos de inconsistência semântica descritos anteriormente.

Como resultado intermediário do processo de adaptação de um documento, é gerada uma lista de eventos representando todas as mídias que devem ser interrompidas e ativadas. Analisando esta lista é possível, portanto, identificar o primeiro tipo de inconsistência descrito (caminhos de links distintos chegando a uma mesma mídia): basta verificar se existe mais de um evento referente a uma mesma mídia. Por exemplo, se existem dois eventos “A:stopped” ou se existe um evento “A:stopped” e um evento “A:started”. Desta forma, a execução de todas as adaptações possíveis e a análise das listas de eventos geradas durante essas adaptações permitem a identificação deste tipo de inconsistência. Para executar todas as adaptações possíveis de uma apresentação deve-se, portanto, simular uma falha de QoS para cada mídia contínua referenciada no documento multimídia.

A simulação de falhas de QoS para executar todas as adaptações possíveis do documento também é uma solução que permite a identificação do segundo tipo de inconsistência semântica (especificação de mídias alternativas jamais ativadas). Após a execução de todas as simulações é possível verificar se algum dos links condicionais especificados não foi disparado em nenhum momento.

A identificação do terceiro tipo de inconsistência (links condicionais desativando mídias já apresentadas) é uma tarefa um pouco mais complexa, uma vez que essa inconsistência depende do instante em que ocorre a falha de QoS em uma mídia e do instante em que a apresentação deve ser reiniciada (definido pelo parâmetro “restart” desta mídia). Neste caso, a simulação de falhas de QoS para uma mesma mídia em instantes distintos da apresentação permite que seja verificado se o processo de adaptação está alterando mídias que já foram apresentadas, o que caracteriza este tipo de inconsistência. Uma solução para determinar em que instantes as simulações devem ser realizadas é a divisão da apresentação em estados, onde cada transição entre os estados é caracterizada pelo início da apresentação de novas mídias e/ou pela finalização de mídias que estão sendo apresentadas. Desta forma, em cada estado tem-se o conhecimento das mídias que já foram apresentadas e das mídias que estão sendo apresentadas.

5.2 - Implementação da Ferramenta SCL Semantics

Basicamente, a ferramenta de verificação semântica realiza os seguintes passos para identificar as inconsistências de um documento adaptativo: (i) a apresentação original (definida no documento SMIL) é dividida em estados; (ii) para cada estado é simulada a falha de QoS de cada mídia contínua e ativa (i.e. que está sendo apresentada) do respectivo estado, resultando na execução de todas as adaptações do documento; (iii) utilizando os resultados das adaptações são identificadas as inconsistências semânticas e as mesmas são apontadas através de uma interface gráfica. No apêndice C são descritas as classes definidas durante a implementação da ferramenta.

A ferramenta inicia o processo de verificação analisando o arquivo SMIL com o objetivo de construir um diagrama de estados representando o escalonamento do documento (*scheduling graph*). Para isso o documento deve estar consistente, ou seja, as restrições temporais de todas as suas mídias devem estar bem definidas. Partindo deste princípio é possível identificar o instante de todos os eventos internos determinísticos da apresentação. Esses eventos podem ser do tipo “s” (start) indicando o início de uma determinada mídia “X”

(e.g. “s_X”), ou do tipo “e” (end) indicando o término de uma determinada mídia “X” (e.g. “e_X”). Eventos externos não- determinísticos, como interações do usuário ou atrasos de rede, não são considerados.

Para a construção do diagrama, os eventos são ordenados cronologicamente e separados em grupos de eventos simultâneos. Cada transição t_i no diagrama de estados é caracterizada por um grupo de eventos simultâneos e possui sua origem em um estado e seu destino em um segundo estado. Por outro lado, cada estado s_i é caracterizado por um conjunto de mídias já apresentadas PM_i (presented media) e um conjunto de mídias ativas AM_i (active media). Se a transição t_i tem origem em s_i e destino em s_{i+1} , sendo ela formada pelo conjunto de eventos simultâneos $[e_{X1} \dots e_{Xj}, s_{Y1} \dots s_{Yk}]$ $j \geq 0, k \geq 0$, e seja $X = \{X1..Xj\}$ e $Y = \{Y1..Yk\}$, então $PM_{i+1} = PM_i \cup X$ e $AM_{i+1} = \{AM_i - X\} \cup Y$. Para o primeiro e último estado (s_0 e s_n) não há nenhuma mídia ativa ($AM_0 = AM_n = \emptyset$). A Figura 5 mostra um exemplo de diagrama de estado representando o escalonamento do timeline ilustrado na mesma figura.

A definição de um diagrama de estados da apresentação permite que a ferramenta analise o comportamento da adaptação em função dos estados em que são simuladas as falhas de QoS. Como foi descrito anteriormente, algumas inconsistências nas informações de adaptação só são identificadas em determinados estados da apresentação. Por exemplo, a substituição de uma mídia A por sua alternativa somente é identificada como uma inconsistência se a adaptação é disparada em um estado s_i onde A não está mais ativa (i.e. $A \notin AM_i$ e $A \in PM_i$).

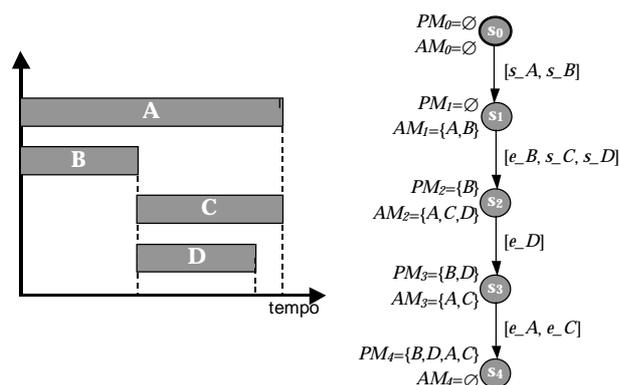


Figura 5 - Diagrama de estados para o escalonamento de uma apresentação

Definido o diagrama de estados, a ferramenta percorre cada estado s_i (com exceção do primeiro e último estado) e simula uma falha de QoS para cada mídia contínua A (áudio, vídeo ou animação) que esteja ativa no estado ($A \in AM_i$). Para cada falha de QoS simulada, um processo de adaptação forte é disparado. Neste ponto é realizada uma chamada ao módulo Adapter. O Adapter executa o processo de adaptação, retornando como resultado a lista de todos eventos disparados (indicando as mídias que são interrompidas e as alternativas que são ativadas) e o documento SMIL adaptado, caso nenhum erro tenha ocorrido durante a adaptação. Esses erros podem ser causados, por exemplo, por problemas na sintaxe do documento SCL ou por referências a mídias inexistentes no documento SMIL.

Para cada documento SMIL adaptado retornado pelo Adapter é construído um novo diagrama de estados representando seu escalonamento. Em seguida, este novo diagrama é agregado ao diagrama da apresentação original. Para ilustrar o processo de agregação dos diagramas de estados, considere o exemplo da Figura 6. Nessa figura é ilustrado o timeline de uma apresentação original (Figura 6.a) e o seu respectivo diagrama de estados (Figura 6.b). Neste exemplo é simulada uma falha de QoS da mídia “C” dentro do estado “sA2”. De acordo com os links condicionais definidos para este documento (Figura 6.c), a apresentação é

adaptada ocorrendo a substituição da mídia “C” pela mídia “F”. A Figura 7 ilustra o timeline da apresentação adaptada (Figura 7.a) e o diagrama de estados gerado para esta apresentação (Figura 7.b).

Supondo que o parâmetro “restart” da mídia “C” tenha o valor “media”, a apresentação adaptada deve ser iniciada a partir do início da mídia “F”. Como essa mídia só inicia sua apresentação em “sB2”, os dois primeiros estados do diagrama da apresentação adaptada podem ser desconsiderados. Portanto, a apresentação adaptada é iniciada, efetivamente, em “sB2” (Figura 7). A agregação entre os dois diagramas deve ser realizada criando-se uma transição para conectar o estado em que é simulada a falha de QoS (“sA2”) e o estado em que a apresentação adaptada é iniciada (“sB2”). A lista de eventos que caracteriza essa transição é definida com base nos conjuntos de mídias ativas desses dois estados. No exemplo, como “MA2={A,C}” e “MB2={A,F}”, a lista de eventos da transição entre “sA2” e “sB2” deve ser “[e_C,s_F]” (Figura 8.a).

Existem ainda outras duas situações em que o processo de agregação pode ser executado: quando “restart=failure” ou “restart=document”. Suponha que seja simulada a falha de QoS de “X”, e que a mídia “Y” substitui “X” durante a adaptação. No caso em que o parâmetro “restart” de “X” possui o valor “failure”, para executar o processo de agregação

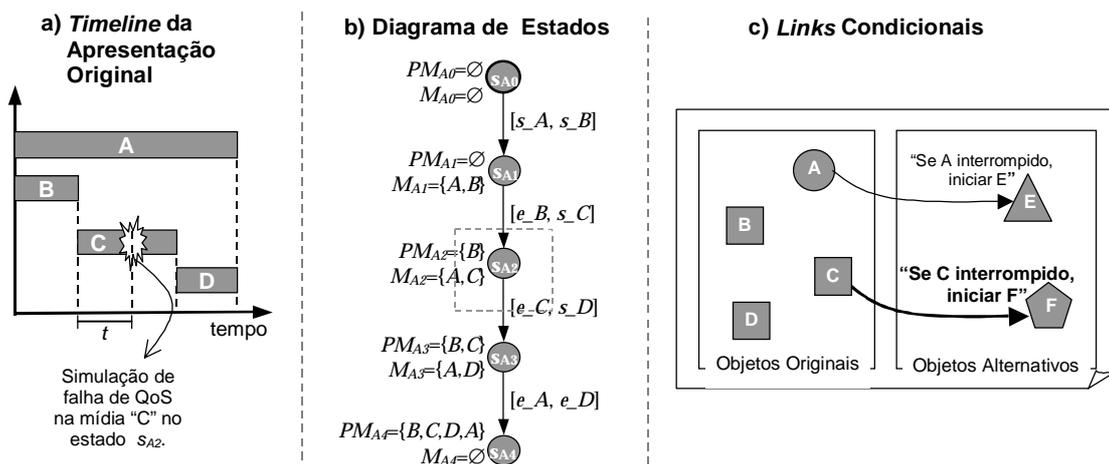


Figura 6 - Apresentação de um documento adaptativo

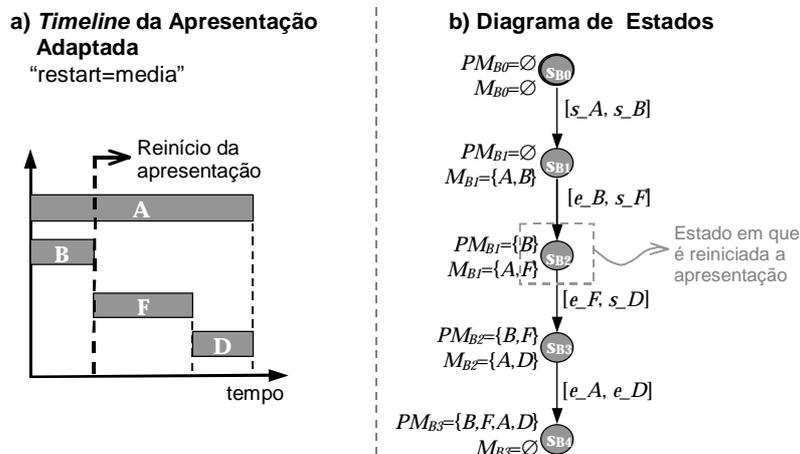


Figura 7 - Apresentação adaptada

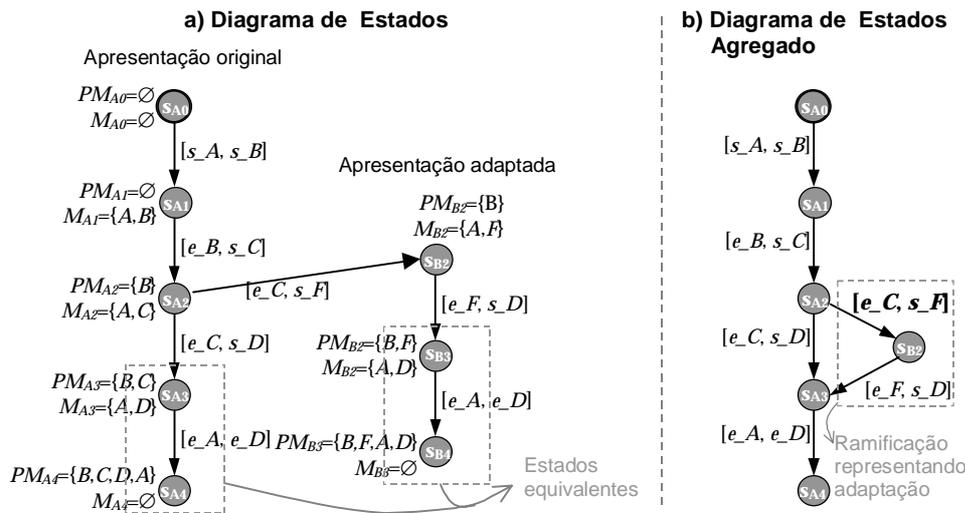


Figura 8 - Agregação dos diagramas de estados

entre os diagramas da apresentação adaptada e da apresentação original é necessário saber quanto tempo de “X” já foi apresentado até o instante em que é simulada a falha de QoS (seja “tX” este tempo). Da mesma forma, dentro do diagrama de estados também devem ser informados os instantes de ocorrência de cada transição (indicando quando ocorre cada evento). Com isso, é possível saber a partir de qual dos estados em que a mídia “Y” encontra-se ativa deve-se iniciar a apresentação adaptada. Neste caso, se “s_Y” ocorre no instante “tsY”, basta identificar qual estado equivale ao instante “tsY+tX”. Se “Y” só está ativa em um estado, a agregação ocorre de forma equivalente ao caso em que “restart=media” (esta situação ocorreria no exemplo anterior se o parâmetro “restart” de “C” fosse igual a “failure”).

No caso em que o parâmetro “restart” tem valor “document”, a agregação entre os dois diagramas é realizada entre o estado da apresentação original em que é simulada a falha de QoS e o segundo estado da apresentação adaptada (o primeiro estado representa a apresentação ainda não iniciada). Se o processo de adaptação não alterar nenhuma mídia na parte da apresentação anterior ao instante em que é simulada a falha de QoS, os estados iniciais dos dois diagramas serão idênticos. Neste caso, pode-se criar uma transição ligando o estado em que ocorre a falha e o segundo estado da própria apresentação original. No entanto, isso pode comprometer a ilustração do comportamento adaptativo da apresentação. Os valores “document” e “failure” do parâmetro “restart” ainda não são suportados na versão atual da ferramenta “SCL Semantics”. Somente o valor default (“media”) está sendo considerado.

Como resultado do processo de agregação, a apresentação adaptada passa a ser representada por uma ramificação partindo do estado em que foi simulada a falha de QoS. Dependendo da apresentação gerada pelo processo de adaptação, esta ramificação pode retornar ao caminho principal (caminho definido pelo diagrama de estados da apresentação original). Isso ocorre quando o processo de adaptação manipula (substitui ou cancela) somente as mídias de um determinado trecho da apresentação. Desta forma, os estados finais da apresentação adaptada são equivalentes ao da apresentação original. Ainda considerando o exemplo anterior, como é apontado na Figura 8.a, os estados finais “sB3” e “sB4” da apresentação adaptada são equivalentes aos estados “sA3” e “sA4” da apresentação original. Esta equivalência permite, portanto, que os estados “sB3” e “sB4” sejam desconsiderados e que o estado subsequente a “sB2” seja “sA3”. O resultado final da agregação dos diagramas de estados das duas apresentações é ilustrado pela Figura 8.b. É importante observar que o

último estado do diagrama de uma apresentação adaptada é sempre equivalente ao último estado do diagrama da apresentação original.

Finalmente, após a realização de todas as adaptações possíveis, a ferramenta gera um diagrama de estados onde estão presentes todas as ramificações representando essas adaptações. Este diagrama permite a visualização de todo o comportamento adaptativo da apresentação. Para cada ramificação de adaptação, a ferramenta apresenta a lista de stoplinks e startlinks disparados, apontando, quando for o caso, dois dos três tipos de inconsistências semânticas descritas anteriormente: (i) ativação e desativação de uma mesma mídia e (ii) substituição de mídias já apresentadas. O terceiro tipo de inconsistência nas informações de adaptação é identificado caso seja verificado que algum dos links condicionais especificados não é disparado em nenhum momento.

5.3 – Interface da Ferramenta SCL Semantics

. Após ter criado os documentos SMIL e SCL, o autor entra com o nome do arquivo SMIL (o arquivo SCL é homônimo, possuindo a extensão “.scl”). Inicialmente a ferramenta exibe apenas o diagrama de estados representando o escalonamento da apresentação especificada pelo documento SMIL.

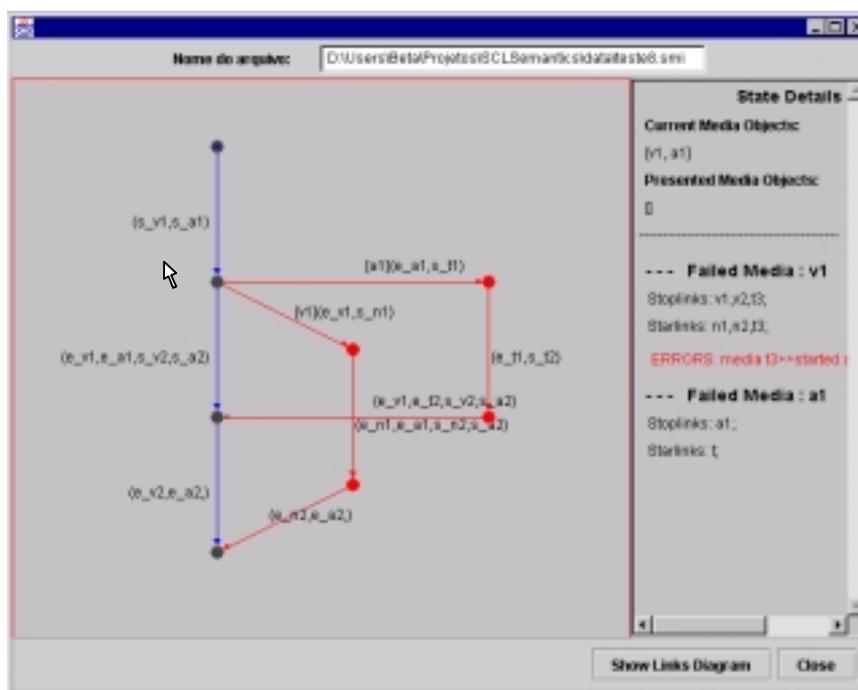


Figura 9 - Ferramenta SCLSemantics: caminhos representando adaptações

Quando um dos estados é clicado, o diagrama é “explodido” exibindo novos caminhos ou ramificações (Figura 9, Figura 10). Como foi dito anteriormente, essas ramificações representam as alterações no comportamento da apresentação devido às possíveis adaptações no documento original. Cada nova transição que parte do estado clicado é rotulada com o “id” da mídia cuja falha de QoS foi simulada (e.g. “[a1]”) e a lista de eventos representando o re-escalamento da apresentação em direção ao primeiro estado da ramificação de adaptação (e.g. “(e_a1,s_t1)”).

A ferramenta SCLSemantics possui um módulo complementar para a visualização de todos os links condicionais especificados no documento SCL. Quando o botão “Show Links Diagram” é clicado, é exibida uma janela (Figura 11) contendo um diagrama ilustrando as mídias originais e alternativas e os links entre elas (startlinks em azul e stoplinks em vermelho). Os links associados às inconsistências identificadas pela ferramenta são destacados.

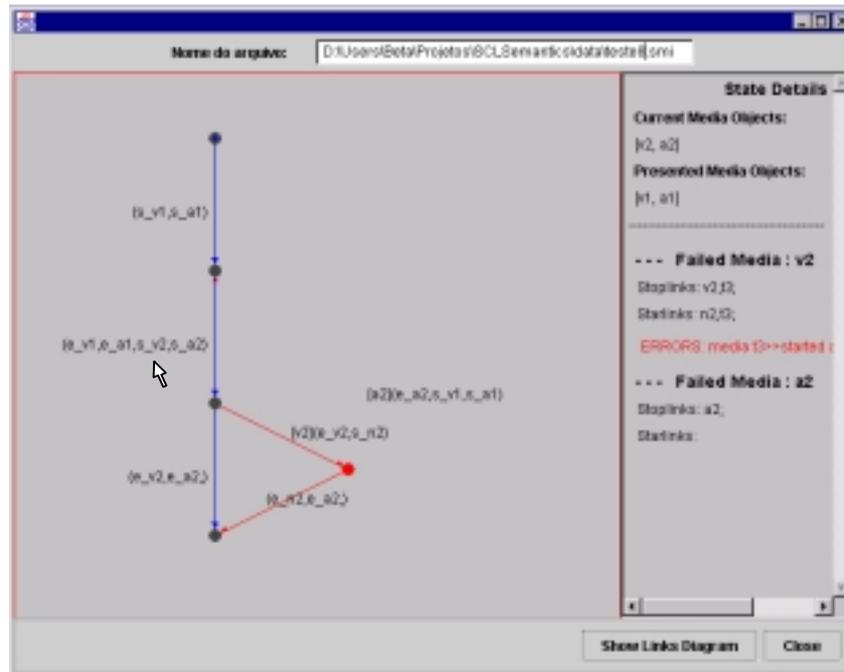


Figura 10 - Ferramenta SCLSemantics: caminhos representando adaptações

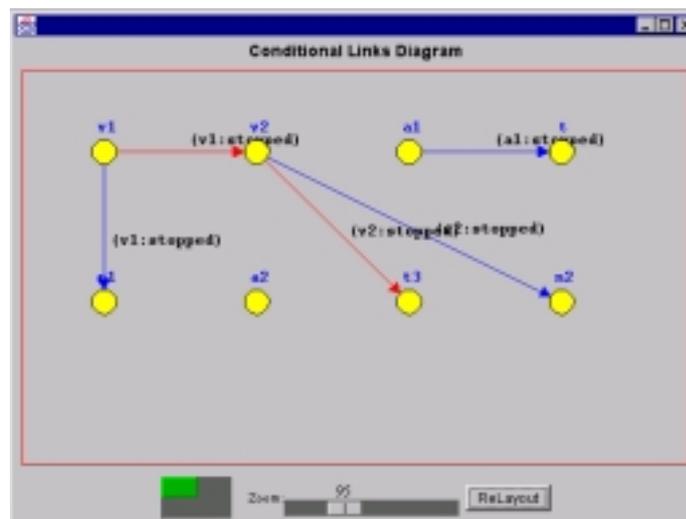


Figura 11 - Módulo complementar (diagrama de links condicionais)

6. CONCLUSÃO

Neste trabalho foi proposta uma estratégia de autoria e apresentação que visa tornar adaptativas as aplicações multimídia. Esta estratégia baseia-se no estabelecimento de uma malha de relacionamentos condicionais que permite que, durante uma apresentação, seja verificada a necessidade de adaptar alguma informação e que define a forma como esta adaptação deve ser realizada.

Durante o trabalho, foi implementado o módulo Adapter com o objetivo de verificar a viabilidade da estratégia de adaptação proposta com respeito à criação de documentos adaptados em tempo real. Sobre essa implementação, foram realizados alguns testes de forma a examinar o comportamento do tempo de resposta do sistema em função do número de links condicionais do documento adaptativo. Com isso, foi verificado que o número de links condicionais exerce uma influência direta no desempenho do mecanismo. Durante a implementação foi utilizada a plataforma Java API for XML Parsing 1.0, desenvolvida pela SUN. O fato de a implementação ser baseada na linguagem Java impôs uma sobrecarga computacional (overhead) considerável. Mas, por outro lado, a interoperabilidade do sistema foi garantida.

Durante o desenvolvimento da estratégia de adaptação, esperava-se que a adaptação suave apresentasse um tempo de resposta bem melhor que o da adaptação forte. Isto porque o atraso introduzido pela adaptação suave é decorrente apenas do processo de chaveamento de fluxos, enquanto o atraso total introduzido pela adaptação forte é composto pelo (i) tempo de criação do documento adaptado mais o (ii) tempo de chaveamento de fluxos. No entanto, de acordo com os testes apresentados em [9], os atrasos introduzidos no nível de chaveamento de fluxos para os dois mecanismos de adaptação são da mesma ordem (em média, 1500ms para a adaptação suave e 2000ms para a adaptação forte). Além disso, os testes realizados com o módulo Adapter mostraram que o tempo de criação do documento adaptado é bem menor (em média, de cinco a dez vezes menor) do que o tempo de chaveamento de fluxos. Desta forma, o atraso total introduzido pela adaptação forte, ao contrário do que se esperava inicialmente, é da mesma ordem que o atraso introduzido pela adaptação suave.

Em um ambiente de ensino à distância o atraso de adaptação tende a ser aceitável, pois as dimensões temporais das mídias acessadas neste tipo de aplicação são da ordem de alguns minutos. Além disso, quando se trata de uma aplicação voltada ao ensino, é melhor a convivência com um pequeno atraso de chaveamento do que a possibilidade de uma interpretação equivocada das informações. Vale ressaltar novamente que a adaptação forte deve ser ativada apenas nas situações em que o mecanismo de adaptação suave não conseguir manter os requisitos de QoS especificados.

Também como parte do trabalho, foi desenvolvida uma ferramenta de auxílio à autoria de documentos adaptativos. Esta ferramenta permite a identificação de possíveis inconsistências semânticas definidas nas informações de controle. A ferramenta, com base em simulações de falha de QoS, executa todas as possíveis adaptações do documento. Através de uma interface gráfica, são apresentados todos os estados que a apresentação pode percorrer, além das inconsistências identificadas. Durante o desenvolvimento deste trabalho, verificou-se que a especificação de todos os links condicionais de uma apresentação para determinar seu comportamento adaptativo pode ser uma tarefa bastante complexa. Desta forma, a ferramenta implementada mostrou-se bastante funcional no sentido de simplificar esta tarefa e garantir a consistência dos dados de adaptação especificados.

REFERÊNCIAS

- [1] CARMO, L.F.R.C., PIRMEZ, L. “ServiMídia: An Integrated System for Multimedia Document Creation and Retrieval with Adaptive QoS Control”. In: FRANCE-BRAZIL SYMPOSIUM ON DISTRIBUTED COMPUTER SYSTEMS, 2, 1997, Recife, Brazil.
- [2] LIU, C. Multimedia Over IP: RSVP, RTP, RTCP, RTSP. Disponível na INTERNET via [www.url:http://www.netlab.ohio-state.edu/~jain/cis788-97/ip_multimedia/index.htm](http://www.netlab.ohio-state.edu/~jain/cis788-97/ip_multimedia/index.htm). Arquivo consultado em 2000.
- [3] COURTIAT, J.P., CARMO, L.F.R.C, OLIVEIRA, R.C. “A General-purpose Multimedia Synchronization Mechanism Based on Causal Relations”. IEEE Journal on Selected Areas in Communications, v.14, n.1, p.185-195, January 1996.
- [4] W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. W3C Recommendation, June 1998. Disponível na INTERNET via [www.url:http://www.w3.org/TR/REC-smil](http://www.w3.org/TR/REC-smil). Arquivo consultado em 2000.
- [5] GOMES, ROBERTA L. “Autoria e Apresentação de Documentos Multimídia Adaptativos em Redes Orientador: Luiz Fernando Rust da Costa Carmo. Rio de Janeiro: UFRJ/IM/NCE, Agosto de 2001. Dissertação. (Mestrado em Informática)
- [6] GOMES, R. L., CUNHA, E. C., CARMO, L. F. R. C., PIRMEZ, L. “An Adaptive Presentation System for Multimedia Documents” In: Seventh International Conference on Intelligent Multimedia and Distance Education, 2001, Fargo. Proceedings of ICIMADE'01 , 2001.
- [7] SCHULZRINNE, H., RAO, A., LANPHIER, R. “Real Time Streaming Protocol (RTSP)”. IETF RFC 2326, abril 1998.
- [8] SCHULZRINNE, H. et al. “RTP: A Transport Protocol for Realtime Applications”. IETF RFC 1889, jan. 1996.
- [9] CUNHA, E.C. Uma Estratégia de Criação e Apresentação de Documentos Multimídia Adaptativos em Rede. Orientador: Luiz Fernando Rust da Costa Carmo. Rio de Janeiro: UFRJ/IM/NCE, 2000. Dissertação. (Mestrado em Informática)