Aplicativo em Tcl/Tk para Composição de Objetos Multimídias

UFRJ

Projeto submetido para a obtenção do título de

Engenheiro Eletrônico

no Departamento de Engenharia Eletrônica

da EE/UFRJ

por

Paulo André da Silva Gonçalves

Aplicativo em Tcl/Tk para Composição de Objetos Multimídias

Paulo André da Silva Gonçalves

Orientadores:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing.

Ronaldo Mussauer de Lima, M.Sc.

Examinadores:

Prof. José Ferreira de Rezende, Dr.

Marcelo Gonçalves Rubinstein, M.Sc.

Departamento de Eletrônica - Escola de Engenharia - UFRJ

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 1998

Aos meus pais.

Agradecimentos

À minha mãe pelo incentivo, pela dedicação, pelo amor e pelo respeito às minhas decisões, ao meu pai pelo reconhecimento das dificuldades e apoio nestes momentos e à minha irmã por sua compreensão e constante ajuda.

Aos meus familiares Arlete, Silas, Rui, Cida, Lucas, Suellen, Almir, Deléia, Kleiton, Ozia, Elaine, Cláudio, Maria, Andres e, principalmente, à Vera (*in memorian*) por acreditarem em minha dignidade e caráter.

Ao Prof. Otto Carlos Muniz Bandeira Duarte por acreditar em meu potencial, orientando e estimulando o desenvolvimento deste trabalho.

Aos professores Otto, Aloysio, Mauros, Casé, Antônio Cláudio e Roosevelt pela contribuição inestimável que deram à minha formação profissional.

Aos meus colegas de turma, principalmente, ao Alexandre e ao Artur por todos os trabalhos que realizamos juntos.

Ao Ronaldo Mussauer por dedicar parte de seu tempo no apoio deste trabalho, mesmo nos momentos de *stress*.

Ao Rubi e ao Baiano pela amizade e companheirismo demonstrados nestes anos de convivência.

Aos membros da banca examinadora (Otto, Ronaldo, Rezende e Rubi) pela presença e contribuições acrescentadas a este trabalho.

À Renata pelos "toques" e incentivos durante a fase de elaboração do Projeto Final.

À equipe GTA. Pessoal do doutorado: Baiano, Rubi, Belem, Bill, Marcos e Vidal. Pessoal do mestrado: Fred, Renata & Flávio, Andrea, LuisH, Daniel, Emerson, Hermes, Antônio, Adriana, Sílvia, Jonny e Spindola.

Aos alunos novos e antigos de Iniciação Científica: Godinho, Bernardo, Tadeu, Daniel (doc) e Gardel.

Aos alunos novos no GTA: Jucele, Freire, Ezequiel e Duboc.

À funcionária Conceição do DEL pela simpatia no atendimento às pessoas e pela dedicação nos serviços prestados.

À Evelize, Fernanda, Kelli & Sérgio, Júlio & Flávia, Rafael, Márcio e Ralph.

À todos que me incentivaram, contribuindo de forma direta ou indireta para minha formação profissional.

Palavras-Chave

Multimídia

Sincronização

Redes de Petri

Relações Temporais

Resumo do Projeto Final apresentado ao Departamento de Eletrônica como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico.

Aplicativo em Tcl/Tk para Composição de Objetos Multimídias

Paulo André da Silva Gonçalves

Março de 1998

Orientadores:
Otto Carlos Muniz Bandeira Duarte
Ronaldo Mussauer de Lima

Uma apresentação multimídia exige a realização de tarefas que envolvam atividades paralelas e seriais. Uma forma de especificar claramente e lidar mais facilmente com as relações temporais entre as mídias de uma apresentação é utilizar um método para modelar estas relações. Assim, para um Sistema Multimídia, é de vital importância a utilização de uma ferramenta, com base em alguma técnica, que permita especificar formalmente a sincronização entre os objetos multimídias e modelar completamente uma apresentação multimídia. A partir desta demanda foi elaborado o aplicativo *Composition* que permite ao usuário especificar sinteticamente a sincronização entre mídias previamente armazenadas na base de dados de um Sistema Multimídia e compor qualquer tipo de apresentação. Este aplicativo utilizou como base o modelo OCPN (*Object Composition Petri Net*) de Little et al. que é um método geral proposto para especificar a sincronização e realizar a composição temporal de objetos multimídias. O aplicativo se mostrou eficiente e seguro para a composição dos objetos multimídias que farão parte da apresentação que se deseja criar.

Sumário

Resumo **v**

I Introdução **1**

I.1 Sistemas Multimídias 3

I.2 Proposta de Trabalho 4

II Conceitos Básicos de Sincronização Multimídia **6**

II.1 Mídias Dependentes e Independentes no Tempo 6

II.2 Sincronização Multimídia 7

II.3 Sincronização Intramídia e Intermídia 9

II.4 LDUs (*Logical Data Units*) 10

II.5 Sincronização Sintética e ao Vivo 12

III Métodos de Especificação da Sincronização Multimídia **14**

III.1 Especificações Baseadas em Intervalos 15

III.2 Especificações Baseadas em Eixos 18

III.3 Especificações Baseadas em Eventos 20

III.4 *Especificações Baseadas em Scripts* 21

III.5 Especificações Baseadas em Controle de Fluxo 22

III.6 O Modelo OCPN 24

IV O Aplicativo **28**

IV.1 Tcl/Tk - O Conjunto de Ferramentas 28

IV.2 Arquivo de Definição da OCPN 31

IV.3 Tabela de Conversão de Objetos 32

IV.4 Arquivo de Dados Adicionais do Objeto 34

IV.5 Módulos Componentes 36

IV.6 Recursos 37

IV.7 Construção de Diagramas 39

IV.8 Atribuição de Propriedades 41

IV.9 Formato dos Arquivos *dgm* 42

V Considerações Finais **46**

Referências Bibliográficas **49**

A Listagem **52**

Lista de Figuras

II.1 Sincronização intramídia entre quadros de uma seqüência de vídeo mostrando uma pequena bola quicando. 9

II.2 Exemplo de sincronização intermídia que mostra relações temporais em uma apresentação multimídia que inclui áudio, vídeo, animação e imagens. 10

II.3 Exemplo de LDUs de vídeo. 10

II.4 Exemplo de LDUs de áudio. 11

II.5 LDUs de duração variável. 11

III.1 Tipos de relações temporais entre dois objetos. 16

III.2 Operações presentes no método de especificação baseado em intervalos. . . 17

III.3 Exemplo de especificação da sincronização baseada em eixos de tempo. . . 19

III.4 Exemplo de aplicação com especificação de sincronização baseada em eixos virtuais. 19

III.5 Apresentações serial e paralela. 23

III.6 Hierarquias em redes de Petri temporizadas. 24

III.7 Diagrama de uma OCPN. 26

III.8 Relacionamentos temporais e OCPNs correspondentes. 27

IV.1 Biblioteca Tcl e alguns comandos específicos do aplicativo. 29

IV.2 Biblioteca Tcl, Biblioteca Tk e dois conjuntos de comandos (pacotes) definidos pelo programador. 30

IV.3 Exemplo da utilização do Tk. 30

IV.4 Diagrama dos principais módulos do ***Composition***. 36

IV.5 Menu *File*. 38

IV.6 Menu *Edit*. 38

IV.7 Menu *Specification*. 38

IV.8 Barra de ferramentas superior. 39

IV.9 Barra de ferramentas lateral. 39

IV.10Diagrama de uma OCPN construído com o ***Composition***. 40

IV.11Janela inicial para a atribuição de propriedades. 41

IV.12Janela de atribuição de propriedades para o recurso vídeo. 42

IV.13Objetos com propriedades definidas. 43

Lista de Tabelas

III.1 Vantagens e desvantagens da especificação da sincronização baseada em intervalos. 18

III.2 Vantagens e desvantagens da especificação da sincronização baseada em eixos de tempo. 20

III.3 Exemplo de especificação baseada em eventos. 21

IV.1 Conversão de Objetos: arquivo *ObjMM*. 33

Capítulo I

Introdução

Com o desenvolvimento e o avanço das pesquisas relacionadas à tecnologia digital e à microeletrônica, a técnica de transmissão digital foi reconhecida pela indústria de telecomunicações como mais eficiente, confiável e econômica do que a analógica. Isto impulsionou o desenvolvimento das redes de alta velocidade, tais como FDDI (*Fiber Distributed Data Interface*), DQDB (*Distributed Queue Data Bus*) e RDSI (*Rede Digital de Serviços Integrados*).

As Redes Digitais de Serviços Integrados - Faixa Larga (RDSI-FL) se propõem a suportar os diferentes tipos de serviços e aplicações, como aplicações multimídias com vários participantes (vídeo-conferência ou conferência multimídia, tele-seminário), visualização científica (reações nucleares, imagens de satélite), visualização médica e tele-diagnóstico, distribuição domiciliar de TV de alta definição (*Hight Definition TeleVision - HDTV*), vídeo-telefonia, tele-educação (vídeo-biblioteca, educação domiciliar distribuída), realidade virtual, trabalho cooperativo (*Computer Supported Cooperative Work - CSCW*) entre outras [1]. Uma tecnologia proposta para suportar a diversidade de serviços, satisfazendo aos requisitos dos diferentes tipos de tráfego, a altas velocidades de transmissão é a ATM (*Asynchronous Transfer Mode*).

O crescente interesse em aplicações multimídias distribuídas, que integram diferentes tipos de mídias, tais como imagens, textos, gráficos, áudio e vídeo [2], incentiva a pesquisa e o desenvolvimento de sistemas que possam dar suporte aos serviços que esta classe de aplicações demanda. Isto inclui a captura, a manipulação, a composição, a integração, o armazenamento, a recuperação, a comunicação e, principalmente, a apresentação sincronizada das mídias [3].

Os diferentes tipos de mídias possuem requisitos específicos de desempenho relativos às restrições de tempo real e podem sofrer atrasos randômicos durante a transmissão e a recuperação de dados. Com isto, a composição temporal de diversas mídias em uma aplicação requer a utilização de técnicas específicas de sincronização que possam prover o suporte à comunicação de dados multimídias e à apresentação coerente no destino. Neste sentido, as técnicas de sincronização utilizam mecanismos para coordenar a ordenação temporal e espacial dos eventos relacionados à aplicação [4].

Nos últimos anos, vários trabalhos de pesquisa em sincronização multimídia têm sido desenvolvidos e publicados. Existem várias propostas com o objetivo de atender aos diversos requisitos de sincronização. Na proposta de Nicolau [5], é apresentada uma arquitetura para sistemas de comunicação multimídias de tempo real, utilizando-se técnicas de sincronização baseadas em protocolos. A proposta de Escobar et al. [6] que também utiliza técnicas baseadas em protocolos, insere estampas de tempo nos dados na fonte e utiliza um atraso de equalização em todos os destinos, fixando um atraso igual para todos os fluxos. Horn e Stefani [7] definem uma linguagem de programação síncrona, denominada ESTEREL, para dar suporte à sincronização de objetos multimídias. Anderson e Homsy [8] descrevem algoritmos para sincronizar dispositivos de entrada (fontes) e saída (destinos) de mídias contínuas, em uma única estação, os quais são conectados a um servidor que utiliza um sistema operacional de tempo real. Ramanathan e Rangan [9] propõem técnicas de realimentação, destinadas à sincronização de sistemas multimídias distribuídos, e protocolos para sincronização intramídia e intermídias de dados pré-armazenados, na presença de variação de atraso da rede e de diferenças não determinísticas nas taxas de captura e/ou apresentação de mídias de diferentes estações. Shepherd et al. [10] descrevem uma plataforma de suporte à comunicações baseadas em objetos para aplicações multimídias distribuídas em redes de alta velocidade. Little et al. [11] propõem um modelo baseado em redes de Petri temporizadas para especificar as relações temporais entre objetos multimídias. Carmo et al. [12] propõem um conjunto de primitivas para serviços de comunicação multimídia. A técnica proposta por Qazi et al. [13] utiliza um modelo estendido da Rede de Petri temporizada para a composição de objetos, para especificar os mecanismos de sincronização e de comunicação de objetos multimídias distribuídos.

I.1 Sistemas Multimídias

Um Sistema Multimidía é qualquer sistema ou aplicação que suporte o processamento integrado de vários tipos de mídias com pelo menos uma mídia dependente no tempo [14].

A origem dos Sistemas Multimidías tem sua raiz na grande evolução da tecnologia digital. Esses sistemas possuem como elemento identificador a integração de fatores que são indispensáveis para que todo o sistema funcione adequadamente. Alguns desses fatores são as altas capacidades de geração, armazenamento, comunicação, manipulação e apresentação de diversas mídias que sejam dependentes ou não no tempo. O fator primordial é a integração que é proporcionada pela representação digital de qualquer dado e pela sincronização entre os vários tipos de mídias e dados.

A palavra sincronização em Sistemas Multimídias refere-se às relações temporais, causais e espaciais entre as mídias. Essas relações são naturalmente definidas durante a captura da mídia. Se o objetivo for apresentá-la da mesma forma como foi originalmente capturada, então este processo é denominado *Sincronização ao Vivo*. Essas relações também podem ser especificadas explicitamente no caso de apresentações compostas por mídias capturadas ou criadas de forma independente. Este processo é denominado *Sincronização Sintética*.

Para a Sincronização Sintética é necessário usar um modelo para a especificação e a manipulação das condições e operações de sincronização. Exemplos comuns de operações semelhantes são a apresentação de mídias em paralelo, a apresentação de mídias uma após a outra (serialmente) e a apresentação de mídias independentes de outras.

Assim, utilizar alguma técnica que permita não só especificar formalmente a sincronização entre os objetos multimídias, como também, modelar a apresentação multimídia por completo é uma necessidade vital para a implantação de um Sistema Multimídia.

I.2 Proposta de Trabalho

O Grupo de Teleinformática e Automação (GTA) da UFRJ vem investido em vários tipos de pesquisas, inclusive às relacionadas com Sistemas Multimídias. São estimulados o desenvolvimento de novas arquiteturas e protocolos de comunicação apropriados para altas velocidades, mecanismos de controle de tráfego de alto desempenho, mecanismos para garantia de qualidade de serviço das aplicações e mecanismos de sincronização.

Na área de Sincronização Multimídia existem vários trabalhos realizados. Lima [1] propõe um serviço de sincronização multimídia que oferece facilidades de suporte à comunicação e apresentação em tempo real de informações multimídias pré-orquestradas, armazenadas em bases de dados remotas. Lima e Duarte [15] implementam o serviço de sincronização multimídia proposto para a comunicação de dados pré-orquetrados. A implementação dos mecanismos de sincronização utiliza o modelo estendido da rede de Petri para a composição de objetos multimídias (XOCPN) [13, 16]. Teixeira [17] complementa o serviço de sincronização contínua para aplicações multimídias proposto por Rubinstein [18]. Rubinstein e Duarte [19] além de definirem um serviço de sincronização contínua para aplicações multimídias, implementam um serviço de sincronização baseado em relações temporais implícitas.

Dando continuidade a estes trabalhos, observou-se a necessidade da automatização de alguns processos relativos ao mecanismo de sincronização. Um destes processos foi automatizado por Junqueira [20] com a implementação de um mecanismo de geração automática da XOCPN para dar suporte ao serviço de sincronização proposto por Lima [1]. Nesta implementação foram propostos algoritmos para o planejamento da recuperação e da transmissão de dados armazenados e para a geração da XOCPN de recepção e transmissão. Foram propostos ainda, um critério de decisão para fluxos paralelos que possuam sincronização intermídia e o seu algoritmo para o modelo XOCPN, que baseou-se na teoria de grafos.

I.2 Proposta de Trabalho

Um outro processo a ser automatizado relacionava-se ao desenvolvimento de uma ferramenta que permitisse ao usuário não só especificar formalmente a sincronização entre os objetos multimídias, como também modelar a apresentação multimídia por completo, gerando assim os dados necessários para a geração automática da XOCPN.

Desta forma, o objetivo deste projeto é, através da utilização do ambiente de programação Tcl/Tk (*Tool Command Language/Toolkit*) [21, 22], elaborar um aplicativo que permita ao usuário especificar facilmente a sincronização entre as mídias previamente armazenadas na base de dados do Sistema Multimídia e montar qualquer tipo de apresentação multimídia. Este aplicativo tem como base o modelo OCPN (*Object Composition Petri Net*) [11, 23, 24, 25, 26] de Little et al. que é um método geral para especificar a sincronização e realizar a composição temporal dos objetos multimídias.

A descrição deste projeto está dividida em cinco capítulos. No capítulo II são apresentados os conceitos e definições básicas com relação aos termos envolvidos na sincronização multimídia. No capítulo III são mostrados alguns métodos de especificação da sincronização multimídia e é apresentado de modo formal o modelo OCPN. O capítulo IV trata do aplicativo desenvolvido. São apresentados o conjunto de ferramentas utilizadas; os tipos de arquivos utilizados pelo aplicativo; os módulos componentes do aplicativo, descrevendo as facilidades proporcionadas ao usuário; a construção do diagrama da OCPN e a atribuição de propriedades aos objetos multimídias. O capítulo V apresenta as conclusões com relação ao trabalho desenvolvido, a influência no desempenho do aplicativo com a utilização do ambiente de programação Tcl/Tk e propostas para trabalhos futuros.

Capítulo II

Conceitos Básicos de Sincronização Multimídia

Neste capítulo são apresentados os conceitos e definições básicas com relação aos termos envolvidos na Sincronização Multimídia. Embora muitas definições possam ser encontradas na literatura, normalmente, elas apresentam pequenas variações.

Para as definições que seguem, o termo **objeto multimídia**, ou simplesmente **objeto** é considerado como sendo uma unidade de informação multimídia, complexa ou simples, sendo constituída por elementos de dados relativos à uma determinada mídia dependente ou não do tempo [1, 27]. Estes termos são amplamente utilizados para referenciar os elementos de dados multimídia. O termo **fluxo** é considerado como sendo uma seqüência temporal de objetos.

II.1 Mídias Dependentes e Independentes no Tempo

As mídias podem ser classificadas como dependentes e independentes no tempo. As mídias dependentes no tempo são aquelas em que as consecutivas unidades que as compõem possuem relações temporais entre si. Se todas essas unidades que as compõem possuem mesma duração de apresentação, elas são denominadas *mídias contínuas*. O vídeo, por exemplo, consiste de um número de quadros ordenados onde cada um deles possui uma duração fixa de apresentação. As mídias independentes no tempo não possuem nenhuma relação temporal na apresentação de seus conteúdos. Como exemplo, existem os textos, gráficos e imagens.

II.2 Sincronização Multimídia

A palavra sincronização é referente ao tempo. Sincronização em Sistemas Multimídias refere-se à relação temporal entre as mídias. De forma mais geral e amplamente usada, o significado da sincronização em Sistemas Multimídias refere-se às relações espaciais, causais e temporais entre as diversas mídias.

Relações de Conteúdo

Relações de conteúdo definem uma dependência da midía com relação a alguns dados. Dois gráficos que são baseados nos mesmos dados mas mostram diferentes interpretações são um exemplo de relação de conteúdo.

Para documentos multimídia integrados, é útil expressar essas relações explicitamente para permitir uma atualização automática de diferentes visões do mesmo dado. Neste caso, somente os dados são editados e, para visualizá-los, são definidas a forma de dependência dos dados e as regras de apresentação. Toda a visualização dos dados gerados é automaticamente atualizada e esses dados não podem ser editados diretamente, ou seja, deve haver uma espécie de gatilho relacionado à geração e visualização dos dados. Esta técnica é, por exemplo, usada em sistemas de banco de dados e pode também ser utilizada por diferentes mídias em um sistema multimídia.

Em geral, a implementação das relações de conteúdo em sistemas multimídias está baseada no uso de estruturas de dados comuns ou interfaces que são usadas para apresentar objetos usando mídias diferentes.

Relações Espaciais

As relações espaciais que são usualmente conhecidas como relações de *layout* definem o espaço para a apresentação da mídia em algum tempo determinado num dispositivo de saída. Se o dispositivo de saída é bidimensional, por exemplo, monitor ou papel, o *layout* especifica a área bidimencional a ser utilizada.

Em aplicações de editoração eletrônica (*desktop-publishing*), isto é usualmente expressado utilizando-se quadros de exibição. Um quadro de exibição é apresentado e um conteúdo é associado a ele. A posição do quadro de exibição pode ser fixada em alguma posição do documento, em alguma posição sobre a página ou pode ser relativa ao posicionamento de outros quadros.

O conceito de quadros também pode ser usado para especificar onde é colocada uma unidade de apresentação de uma mídia dependente no tempo. Por exemplo, quadros de vídeo podem ser posicionados usando-se quadros de exibição. Em sistemas orientados a janelas, um quadro ou um grupo de quadros podem ser representados por uma janela. Uma janela pode ter seu tamanho alterado, ser movida, "iconizada", dando adicionais liberdades ao usuário para adaptar a apresentação as suas necessidades.

Dispositivos de saída tridimensionais como experimentos holográficos e projeção tridimensional permitem ao usuário criar apresentações em três dimensões. Em sistemas de janelas, a terceira dimensão é expressada somente em termos de janelas sobrepostas. Dispositivos de saída de áudio estéreo também possuem formas que definem o posicionamento de uma fonte de áudio em uma apresentação. Isto é, por exemplo, usado em áudio e vídeo-conferências para dar ao participante a impressão de uma disposição de cadeiras que está relacionada com a disposição de quadros ou vídeos de outros participantes da conferência. Isto fornece ao usuário uma impressão natural de comunicação, fazendo ele seguir facilmente uma discussão e, portanto, aumenta a aceitação do usuário.

Relações Temporais

Relações temporais definem as dependências temporais entre mídias. Elas são de interesse sempre que existem mídias dependentes no tempo. Um exemplo de relação temporal é a relação entre um vídeo e um áudio que são gravados durante uma reportagem. Para esta aplicação, a relação temporal durante a apresentação das duas mídias deve corresponder à relação temporal no momento da gravação. Isso significa, por exemplo, que os movimentos labiais do repórter devem estar sincronizados com o áudio.

II.3 Sincronização Intramídia e Intermídia

Existem distinções relativas às relações no tempo entre unidades de uma mesma mídia e às relações no tempo entre diferentes mídias. Esta separação ajuda a esclarecer os mecanismos que suportam ambos tipos de relações.

Sincronização intramídia ou intrafluxo refere-se à relação temporal entre várias unidades de apresentação de uma única mídia dependente no tempo. Um exemplo é a relação temporal entre os quadros de uma seqüência de vídeo. Para um vídeo com uma taxa de 25 quadros por segundo, cada um dos quadros deve ser mostrado por um período de 40 ms. A figura II.1 mostra uma seqüência de vídeo para uma pequena bola em movimento.

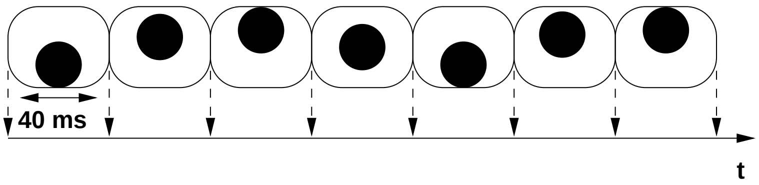

Figura II.1: Sincronização intramídia entre quadros de uma seqüência de vídeo mostrando uma pequena bola quicando.

Sincronização intermídia trata da sincronização entre mídias. A figura II.2 mostra um exemplo das relações temporais de uma sincronização multimídia que inicia com uma seqüência de áudio e vídeo, seguida por diversas imagens e uma animação que é comentada por uma seqüência de áudio.

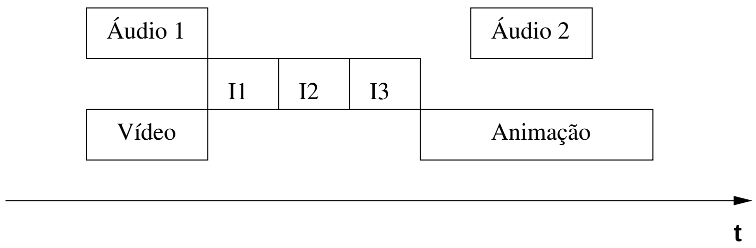

Figura II.2: Exemplo de sincronização intermídia que mostra relações temporais em uma apresentação multimídia que inclui áudio, vídeo, animação e imagens.

II.4 LDUs (*Logical Data Units*)

Mídias dependentes no tempo normalmente consistem de uma seqüência de unidades de informação. Tais unidades de informação são conhecidas como *Logical Data Units* (LDUs).

As LDUs podem ser classificadas em abertas ou fechadas. As LDUs fechadas possuem uma duração prevista, por exemplo, LDUs que fazem parte de uma mídia contínua armazenada, como áudio e vídeo, ou mídias armazenadas que possuam uma duração fixa. LDUs abertas tipicamente representam entradas de uma fonte ao vivo como, por exemplo, uma câmera de vídeo ou um microfone, ou mídias que incluem uma interação com o usuário.

Outro aspecto importante é que uma LDU pode possuir vários níveis de granulidade. Para um vídeo digital, muitas vezes os quadros são selecionados como LDUs. Por exemplo, para um vídeo com uma taxa de amostragem de 30 quadros por segundo, cada LDU é uma LDU fechada com duração de 1/30 segundos (Figura II.3).

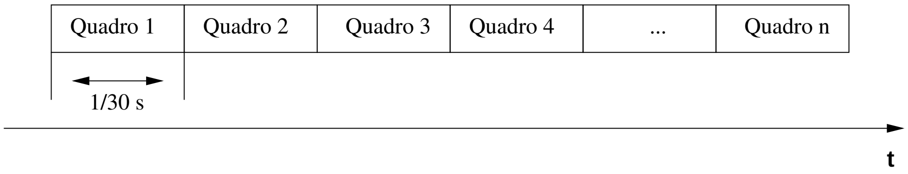

Figura II.3: Exemplo de LDUs de vídeo.

Nos casos em que a unidade física básica é muito pequena, pode-se selecionar LDUs em blocos de amostras dentro de unidades de duração fixa. Um exemplo típico é um áudio onde a duração da unidade física é muito pequena. Neste caso, as LDUs podem ser formadas por 512 amostras. No exemplo da figura II.4, uma amostra é codificada por um octeto, portanto, cada bloco contém 512 octetos.

As mídias capturadas normalmente possuem uma duração básica natural para a LDU. Na geração de mídias por computador, a duração das LDUs pode ser selecionada pelo usuário.

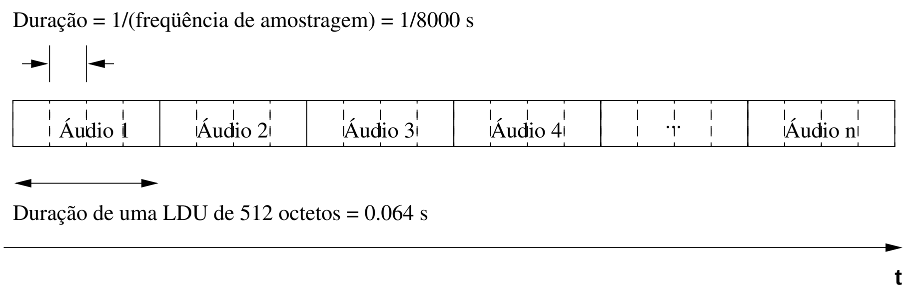

Figura II.4: Exemplo de LDUs de áudio.

Existem alguns casos mais complexos onde a duração da LDU é variável. Um exemplo é a gravação de eventos gerados a partir da interação do usuário com a interface gráfica para uma posterior reapresentação. Neste caso, uma LDU é um evento com uma duração permanente até o próximo evento (Figura II.5).

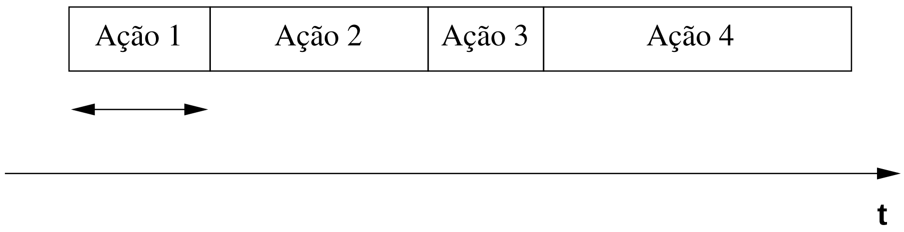

Figura II.5: LDUs de duração variável.

II.5 Sincronização Sintética e ao Vivo

Sincronização sintética e ao vivo referem-se à forma de se determinar relações temporais. No caso da sincronização ao vivo, as relações temporais aparecem implícitas no instante da captura das mídias (gravação das mídias). No caso da sincronização sintética, as relações temporais são sinteticamente especificadas.

Uma aplicação típica da sincronização ao vivo é o serviço de conversação envolvendo a transmissão de áudio e vídeo. Na fonte, os dados são capturados do ambiente e depois apresentados pelo destino. O contexto comum de vários fluxos no local fonte deve ser preservado no destino. Além de sensores acústicos e ópticos, a fonte pode possuir também unidades de conversão de mídias. A fonte e o destino podem estar localizados em lugares diferentes.

Na sincronização sintética, relações temporais são designadas a objetos multimídias que foram criados de forma independente dos outros. A sincronização sintética é freqüentemente usada em apresentações e sistemas de busca de dados com objetos multimídias armazenados que são arranjados para formar novos objetos multimídias. Por exemplo, o mesmo *vídeo-clip* sobre o Brasil pode ser parte de um objeto multimídia que apresenta os países da União Européia, como também pode ser parte de um objeto multimídia que apresenta os países classificados para copa do mundo. Os objetos das mídias envolvidas podem estar armazenados ou localizados em diferentes servidores.

Como já exposto anteriormente, na sincronização sintética é necessário usar um modelo para a especificação e manipulação das condições de sincronização e operações. Exemplos destes tipos de são:

- apresentação de mídias em paralelo;
- apresentação de mídias uma após a outra (serial);
- apresentação de mídias independentes de outras.

A ênfase da sincronização sintética é no suporte de relações de sincronização entre as mídias, de forma flexível, para o qual podemos distinguir duas fases:

- a fase de especificação, na qual são definidas as relações temporais entre as mídias;
- a fase de apresentação, na qual um sistema em tempo de execução apresenta os dados de modo sincronizado.

Capítulo III

Métodos de Especificação da Sincronização Multimídia

A especificação da sincronização entre objetos multimídias deve descrever todas as dependências temporais dos objetos que compõem a apresentação multimídia. Ela pode ser feita utilizando-se alguma ferramenta na camada de aplicação. Como a especificação da sincronização determina toda a apresentação, ela é a característica central dos sistemas multimídias.

Para a complexa especificação da sincronização de múltiplos objetos, incluindo a interação com o usuário, métodos sofisticados de especificação podem ser usados. Os seguintes requisitos devem ser satisfeitos pelo método de especificação [14]:

- o método deve suportar objetos compatíveis e manter as especificações da sincronização. Objetos de uma mídia devem ser mantidos como uma unidade lógica na especificação;
- o método deve suprir uma abstração do conteúdo de uma mídia, permitindo a especificação de relações temporais que se referem a partes da mídia e, por outro lado, considerando a mídia como uma unidade lógica;
- todos os tipos de relações de sincronização devem ser facilmente descritos;

- a integração de mídias dependentes e independentes no tempo deve ser suportada;
- a especificação dos requisitos de QoS (*Quality of Service*) deve ser suportada pelo método. Esta especificação deverá ser expressa preferivelmente no método direto;
- níveis hierárquicos de sincronização devem ser suportados para permitir o manuseio de amplos e complexos esquemas de sincronização.

Muitos métodos de especificação da sincronização têm sido descritos na literatura. Os mais importantes são apresentados neste capítulo. Ao final do capítulo é apresentado formalmente o modelo OCPN que é um método de especificação baseado em controle de fluxo.

Os métodos de especificação de sincronização podem ser classificados nas seguintes categorias :

- especificações baseadas em intervalos, que permitem a especificação das relações temporais entre os intervalos de tempo das apresentações dos objetos multimídias;
- especificações baseadas em eixos, que relacionam eventos da apresentação a eixos que são compartilhados pelos objetos que compõem a apresentação;
- especificações baseadas em eventos, nas quais ações são inicializadas por eventos de sincronização;
- especificações baseadas em *scripts*, nas quais o cenário de sincronização é descrito de modo textual;
- especificações baseadas em controle de fluxo, nas quais existem pontos onde o fluxo da apresentação corrente é sincronizado.

III.1 Especificações Baseadas em Intervalos

Na especificação baseada em intervalos, a duração da apresentação de um objeto está relacionada com um intervalo de tempo. Dois intervalos de tempo podem ser sincronizados de 13 formas diferentes. Algumas destas formas são inversíveis com relação ao intervalo que vem antes ou depois. A figura III.1 mostra um conjunto reduzido de sete tipos não inversíveis [11]. Um método simples de especificação da sincronização para duas mídias é utilizado nestes exemplos.

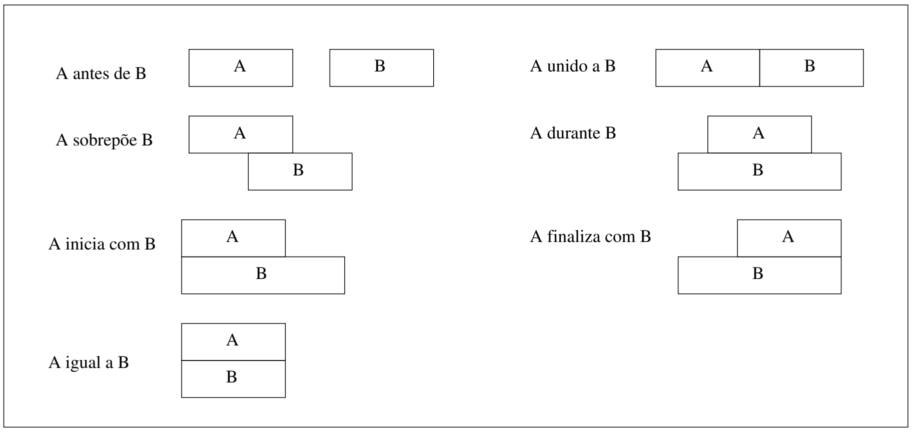

Figura III.1: Tipos de relações temporais entre dois objetos.

O programa a seguir demonstra a utilização do métodos de especificação baseado em intervalos com base no exemplo da apresentação da figura II.2.

```
Audio1 while(0,0) Video
I1 before(0) I2
I2 before(0) I3
I3 before(0) Animation
Animation while(2,5) Audio2
```

Os comandos podem ser entendidos de acordo com a figura III.2.

Este método permite a definição de uma duração para mídias dependentes e independentes no tempo. Esta duração é usada no exemplo para especificar a duração da apresentação das imagens I1 a I3.

A vantagem deste método é a facilidade no manuseio LDUs abertas e, portanto nas interações do usuário. É possível especificar relações temporais não-determinísticas definindo intervalos para durações e atrasos. O modelo não inclui especificações de desvio da

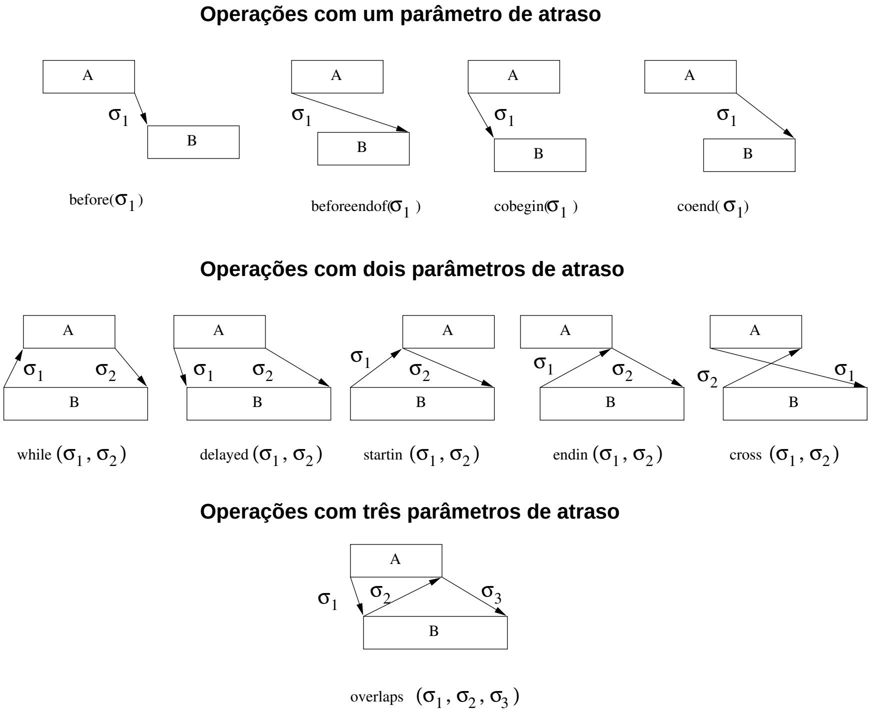

Figura III.2: Operações presentes no método de especificação baseado em intervalos.

QoS. Apesar da especificação direta das relações temporais entre os objetos das mídias, não é permitida a especificação de relações temporais entre sub-unidades dos objetos. Relações semelhantes devem ser definidas indiretamente através da especificação de atrasos.

Um resumo de algumas vantagens e desvantagens deste método é apresentado na tabela III.1 [14].

Tabela III.1: Vantagens e desvantagens da especificação da sincronização baseada em intervalos.

Vantagens	**Desvantagens**
Objetos lógicos podem ser conservados.	Especificação complexa.
Boa abstração para o conteúdo da mídia.	Especificações adicionais necessárias de desvio da QoS.
Fácil integração de objetos de mídias independentes no tempo.	Especificação direta das relações temporais entre objetos da mídia, mas não para subunidades desses objetos.
Fácil integração de objetos interativos.	Resolução do não-determinismo em tempo de execução pode levar a inconsistências.
Especificação de relações temporais não-determinísticas.	

III.2 Especificações Baseadas em Eixos

Na especificação baseada em eixos, eventos da apresentação como iniciar e terminar são mapeados por eixos que são repartidos pelos objetos que compõem a apresentação multimídia.

Sincronização Baseada em um Temporizador Global

Com um temporizador global, a sincronização é descrita através da fixação de todos os objetos em um eixo de tempo, independentes de cada um dos outros. Este eixo representa uma abstração do tempo real.

A figura III.3 mostra a especificação de uma apresentação. Neste caso, não existe nenhum modo natural de lidar com interações de um usuário.

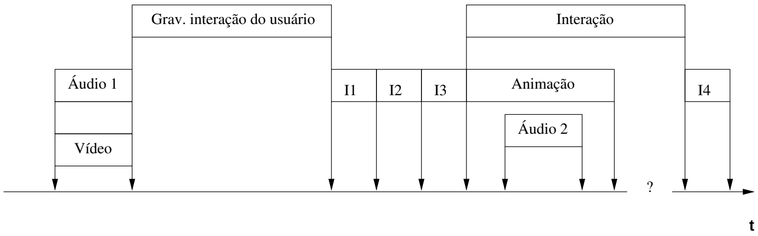

Figura III.3: Exemplo de especificação da sincronização baseada em eixos de tempo.

A tabela III.2 apresenta resumidamente as vantagens e desvantagens deste método [14].

Sincronização Baseada em Eixos Virtuais

A sincronização baseada em eixos virtuais é uma generalização da abordagem da sincronização baseada em um temporizador global. Neste método é possível especificar um sistema de coordenadas com interacões do usuário. A figura III.4 apresenta uma aplicação especificada através deste método.

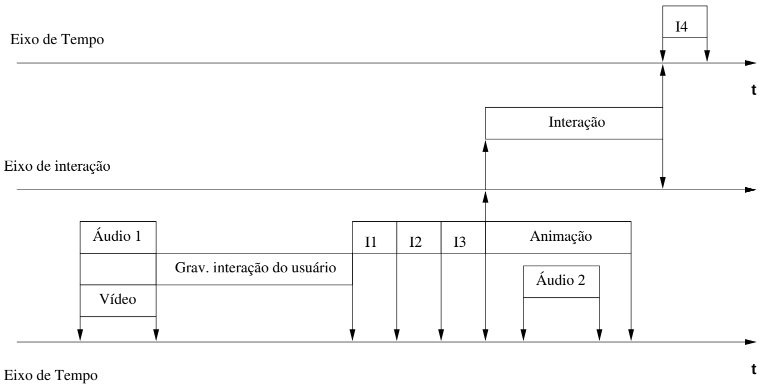

Figura III.4: Exemplo de aplicação com especificação de sincronização baseada em eixos virtuais.

Quando a especificação da apresentação multimídia necessitar de muitos eixos, este método pode ser tornar complexo e o mapeamento dos eixos em tempo de execução pode consumir bastante tempo. Este método porém, é de fácil compreensão, apresenta uma boa

Tabela III.2: Vantagens e desvantagens da especificação da sincronização baseada em eixos de tempo.

Vantagens	**Desvantagens**
Fácil de compreender.	Objetos de duração desconhecida podem não ser integrados. Extensões para o modelo são necessárias.
Suporte de hierarquias fácil de realizar.	Desvio de QoS deve ser especificado indiretamente pelo uso de eixos de tempo comuns ou adicionais especificações de QoS devem ser dadas.
Fácil de manter por causa da independência mútua dos objetos.	
Boa abstração para o conteúdo das mídias.	
Fácil integração de objetos de mídias independentes.	

abstração do conteúdo das mídias, integra facilmente objetos multimídias independentes no tempo e permite que objetos interativos sejam incluídos na apresentação multimídia através de eixos especializados.

III.3 Especificações Baseadas em Eventos

No caso da sincronização baseada em eventos, as ações de uma apresentação são iniciadas por eventos de sincronização. Tipicamente podemos ter as seguintes ações: iniciar, parar e preparar uma apresentação. Um exemplo da especificação baseada em eventos é mostrado na tabela III.3.

Tabela III.3: Exemplo de especificação baseada em eventos.

Evento / **Ação**	**Iniciar**	**Áudio1.parar**	**Temporizador1.pronto**	**...**
Áudio1	Iniciar			
Vídeo	Iniciar			
Imagem1		Iniciar	Parar	
Temporizador1		Iniciar(3)		
Imagem2			Iniciar	
⋮				

Algumas vantagens deste método são: fácil integração de objetos interativos, fácil extensão através de novos eventos e flexibilidade, pois qualquer evento pode ser especificado. Algumas desvantagens são: dificuldade de uso de hierarquias e a necessidade de uma descrição separada dos desvios de QoS. A maior desvantagem deste tipo de especificação é a dificuldade de lidar com cenários mais realísticos onde todos os possíveis eventos não podem ser conhecidos.

III.4 *Especificações Baseadas em Scripts*

Um *script*, no contexto atual, é uma descrição textual do cenário de sincronização. Para isto, o *script* deve proporcionar formas de representar apresentações em série, apresentações em paralelo e repetidas apresentações de uma mesma mídia.

O *script* seguinte exemplifica estas questões. O sinal >> denota uma apresentação em série, & denota uma apresentação em paralelo e **n** denota uma apresentação repetida **n** vezes [14].

```
activity DigAudio Audio("video.au");
```

```
activity SMP Video("video.smp");
activity XRecorder Recorder("window.rec");
activity Picture Picture1("picture1.jpeg");
activity Picture Picture2("picture2.jpeg");
activity Picture Picture3("picture3.jpeg");
activity Picture Picture4("picture4.jpeg");
activity StartInteraction Selection;
activity DigAudio AniAudio("animation.au");
activity RTAnima Animation("animation.ani");
script Picture_sequence 3Picture = Picture1.Duration(5) >>
Picture2.Duration(5) >>
Picture3.Duration(5);
script Lipsynch AV = Audio & Video;
script AniComment AA = Animation & AniAudio.Translate(2);
script Multimedia Application_example {
AV >>
Record.UI >>
3Pictures >>
( (Selection >> Picture4) &
 AA)
```

A utilização deste método para a especificação da sincronização entre os objetos multimídias pode se tornar complexa e difícil de manusear quando a apresentação multimídia for extensa.

III.5 Especificações Baseadas em Controle de Fluxo

Na sincronização baseada em controle de fluxos, o fluxo da apresentação corrente é sincronizado em pontos pré-definidos.

III.5 Especificações Baseadas em Controle de Fluxo

Especificação Hierárquica Básica

A descrição da sincronização hierárquica é baseada em duas principais operações de sincronização: sincronização serial e paralela de ações (Figura III.5). Em uma especificação de sincronzação hierárquica, os objetos multimídias são considerados uma árvore consistindo de nós que denotam apresentações em série ou em paralelo que partem para sub-árvores.

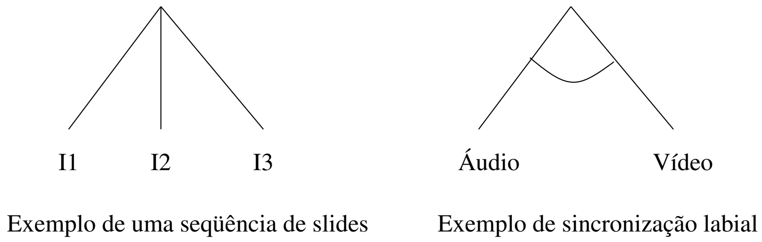

Figura III.5: Apresentações serial e paralela.

Os principais problemas deste método são:

- cada ação só pode ser sincronizada no seu início ou no seu fim. Isto significa que, por exemplo, a apresentação de subtítulos em partes de um fluxo de vídeo requer que este fluxo seja quebrado em vários componentes consecutivos;
- descrição adicional do desvio de QoS é necessária;
- não possui adequada abstração do conteúdo das mídias;
- alguns cenários de apresentação não podem ser descritos.

Redes de Petri Temporizadas

Outro tipo de especificação é baseado em redes de Petri estendidas [24, 25] como será visto na próxima seção.

Para objetos multimídias de mídias dependentes no tempo, cada lugar da rede de Petri representa uma LDU. A sincronização das LDUs é realizada através de suas ligações com as transições.

É possível criar hierarquias através da construção de subredes que são associadas a um determinado lugar. A duração do lugar em questão é associada à duração total da subrede (Figura III.6).

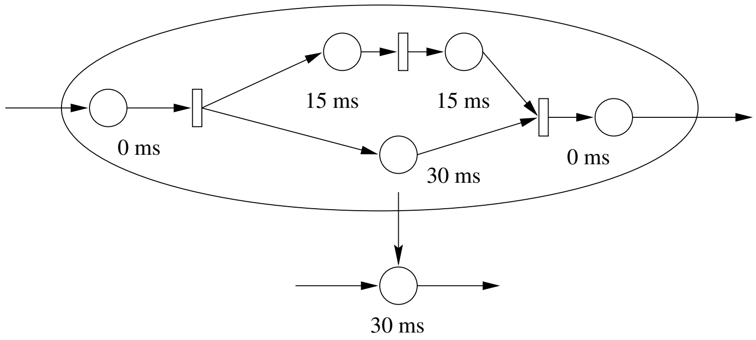

Figura III.6: Hierarquias em redes de Petri temporizadas.

As redes de Petri temporizadas permitem toda forma de sincronização. Outras vantagens dos métodos baseados em redes de Petri estendidas são: a facilidade de criar hierarquias, a fácil integração de mídias independentes no tempo e de objetos interativos e, principalmente, o fato de não existir a necessidade de uma descrição adicional do desvio de QoS, pois a descrição já está integrada ao modelo.

III.6 O Modelo OCPN

O modelo OCPN (*Object Composition Petri Net*) proposto por Little et al. [11] é uma técnica que permite a especificação formal e a modelagem de uma composição multimídia. Este modelo foi elaborado a partir dos modelos de redes de Petri temporizadas, pelo fato destes possuírem atributos de representação de eventos concorrentes e assíncronos.

A rede de Petri é definida como um grafo direcionado bipartite N. Ela é especificada pela tupla {T, P, A} onde :

T = $\{t_1, t_2, ..., t_n\}$ é um conjunto de transições;

P = $\{p_1, p_2, ..., p_m\}$ é um conjunto de lugares;

A = {TxP} U {PxT} $\rightarrow$ I, I = {1, 2,...} é um conjunto de arcos diretos;

Uma rede de Petri com marcação (N_M) = {T, P, A, M} inclui uma marcação M que transfere fichas para cada lugar da rede. Essa marcação é definida como :

> M: P $\rightarrow$ I, I = {0, 1, 2,...} que é um mapeamento a partir do conjunto de lugares para o conjunto de inteiros, que representam o número de fichas em um lugar especificado.

Nas redes de Petri simples, o tempo para habilitar o disparo de uma transição é desconhecido e indeterminado. As redes de Petri temporizadas incluem tempo de duração para transições, e as estendidas representam processos por lugares, ao invés de transições, atribuindo um tempo de execução diferente de zero para os lugares e disparo instantâneo para as transições [1].

Através da OCPN, é possível especificar as diversas relações temporais entre objetos multimídias [25, 24]. Ela é definida como um grafo direcionado bipartite N_{OCPN}. Um grafo G(V, E), que é um grafo G com V vértices e E arestas, é bipartite se e somente se todo o ciclo de G possuir comprimento par [28].

A OCPN é especificada pela tupla {T, P, A, D, R, M}, onde adicionalmente:

> D: P $\rightarrow$ R é um mapeamento a partir do conjunto de lugares para os números reais (durações);
>
> R: P $\rightarrow$ $\{r_1, r_2, ..., r_k\}$ é um mapeamento a partir do conjunto de lugares para um conjunto de recursos;

Associado com a definição da rede de Petri , existe um conjunto de regras de disparo governando a semântica do modelo [25], como se segue:

1. uma transição t_i dispara imediatamente quando cada um dos lugares de origem contém uma ficha acessível;

2. após o disparo, a transição remove uma ficha de cada um dos lugares de origem e adiciona uma ficha a cada um dos lugares de destino;

3. ápos receber uma ficha, um lugar p_j permanece no estado ativo pelo intervalo especificado através da duração τ_j . Durante este intervalo, a ficha está inacessível. Quando o lugar passa para o estado inativo ou ao expirar a duração τ_j , a ficha se torna acessível.

A figura III.7 mostra o diagrama de uma OCPN. A apresentação modelada possui uma duração total de 30 segundos. Em cada lugar do diagrama estão representados o tipo de objeto e a sua respectiva duração em segundos.

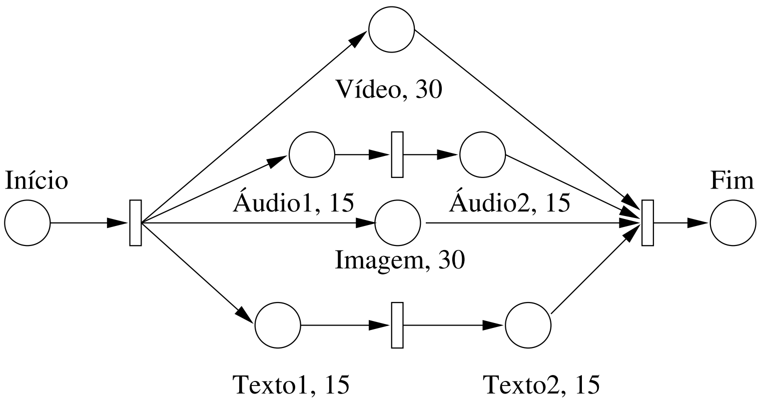

Figura III.7: Diagrama de uma OCPN.

Como apresentado na seção III.1 a duração da apresentação de um objeto está relacionada com um intervalo de tempo. Dois intervalos de tempo podem ser sincronizados de 13 formas diferentes. Alguns desses tipos são inversíveis com relação ao intervalo que vem antes ou depois. Porém, existe um reduzido conjunto de sete tipos não inversíveis.

A figura III.8 mostra como estes relacionamentos são representados pelo modelo OCPN.

A introdução do atraso, de duracão T_σ, como uma ação possível permite então a modelagem de adicionais comportamentos como o atraso em apresentações seriais e paralelas.

O enfoque da OCPN está na sincronização intermídia, onde cada unidade que forma o objeto multimídia é associada com um intervalo de tempo que descreve sua duração. No modelo estendido XOCPN [13], o intervalo de tempo associado é dividido em uma seqüência de unidades menores denominadas SIUs (*Synchronization Interval Units*). Cada

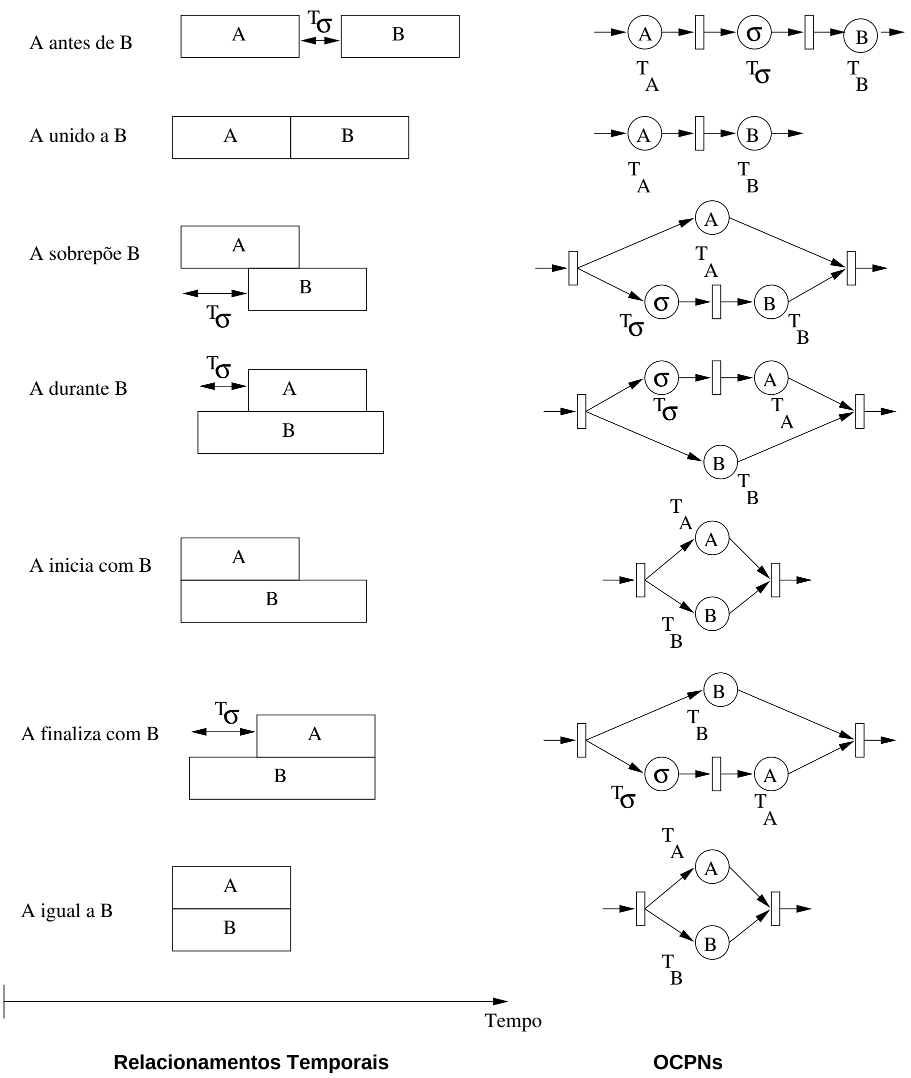

Figura III.8: Relacionamentos temporais e OCPNs correspondentes.

SIU referencia somente parte do objeto e possui tempo de duração igual a das demais.

O principal objetivo da divisão dos objetos em sub-objetos é para melhor coordenar e manter as relações temporais contínuas entre as diversas mídias [1]. Para isto, utiliza-se os dados obtidos a partir da OCPN. A divisão dos objetos também aumenta a eficiência da sincronização intramídia, para o caso de dados contínuos, e permite a sincronização intermídia, para objetos que possuem apresentação concorrente [20].

Capítulo IV

O Aplicativo

O aplicativo construído com o intuito de permitir que o autor da apresentação multimídia (usuário) especifique formalmente a sincronização entre os objetos multimídias e modele a apresentação por completo é o ***Composition***.

O ***Composition*** possui como base o modelo OCPN descrito no capítulo anterior. Seu principal atrativo é o fato de oferecer um ambiente gráfico e simples para que autores de apresentações multimídias possam realizar de forma ágil e segura a composição dos objetos que se deseja criar.

IV.1 Tcl/Tk - O Conjunto de Ferramentas

Como já abordado, foi utilizado o conjunto de ferramentas Tcl/Tk (*Tool Command Language/Tk*) para a construção do aplicativo ***Composition***. O Tcl é uma linguagem de *scripts* para construção de aplicativos. Ela permite a execução de outros programas, além de fornecer recursos para programação como manipulação de variáveis, controle de fluxo, procedimentos, controle de processos, entre outros. Seu interpretador é uma biblioteca de procedimentos C que pode ser facilmente incorporada dentro de aplicativos, e cada aplicativo pode estender o conjunto de comandos definidos no núcleo Tcl através da adição de códigos de programação escritos em C (Figura IV.1).

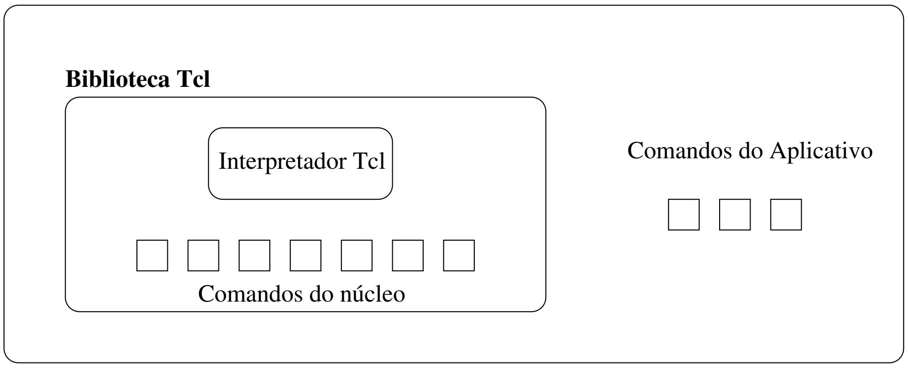

Figura IV.1: Biblioteca Tcl e alguns comandos específicos do aplicativo.

As variáveis definidas no *script* Tcl podem ser "linkadas" com variáveis definidas em procedimentos C adicionados ao núcleo, isto é, toda vez que a variável Tcl mudar de valor, sua correspondente variável C muda também, e vice-versa.

O Tk é uma ferramenta de desenvolvimento de interfaces com o usuário em ambientes *X Windows* que oferece módulos que facilitam a criação de janelas, botões, menus, interação de dispositivos, mensagens de erro, comandos e muitos outros elementos de um ambiente interativo. De forma semelhante ao Tcl, o Tk é um conjunto de bibliotecas C. Por ser uma extensão do Tcl, o Tk, além de possuir seus próprios comandos, incorpora os comandos definidos no núcleo do Tcl. Sendo assim, o programador pode criar interfaces, dispor dos vários recursos de programação acima citados, podendo utilizar também seus próprios códigos escritos em C que foram previamente compilados e adicionados ao núcleo do Tcl (Figura IV.2).

Outra característica do Tk é que o programador pode criar novos tipos de janelas, que possuem aparência e comportamento particulares, através da interface Tk com a linguagem C. Um exemplo simples de utilização do Tk é mostrado a seguir. O *script* cria e mostra um botão com o texto "Sair"que ao ser pressionado executa o comando *exit* que termina a aplicação.

```
button .b -text "Sair" -command exit
pack .b
```

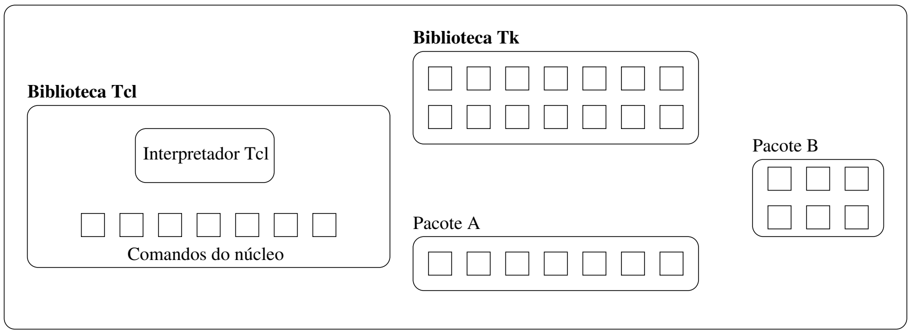

Figura IV.2: Biblioteca Tcl, Biblioteca Tk e dois conjuntos de comandos (pacotes) definidos pelo programador.

O comando *button* somente cria o botão denominado **.b** definindo o texto que irá aparecer e o comando a ser executado. Sua criação não implica em sua visualização na tela. Para que isso ocorra é necessário utilizar o comando *pack* que recebe como parâmetro o nome do botão (*.b*). O resultado da execução deste *script* é mostrado na figura IV.3.

Figura IV.3: Exemplo da utilização do Tk.

Sendo linguagens interpretadas, é possível desenvolver aplicativos de forma independente do sistema operacional e da plataforma utilizada, bastando para isso possuir um interpretador de comandos Tcl/Tk instalado.

IV.2 Arquivo de Definição da OCPN

O serviço proposto em [1] utiliza o modelo OCPN como base para a geração da versão estendida XOCPN. Em sua proposta, o arquivo de definição de uma OCPN é um arquivo ASCII, que permite a descrição de transições e lugares do diagrama da OCPN. Os arcos diretos são implicitamente descritos nos campos das transições e lugares.

A descrição de um lugar deve conter as seguintes informações:

- *Tipo do Componente*: identifica a descrição como sendo a de um lugar;
- *Tipo do Lugar*: identifica o tipo de lugar que pode ser INICIAL, OBJETO ou FINAL;
- *Identificação do Lugar*: número de identificação do lugar da rede em questão;
- *Número do Objeto*: número de identificação do objeto que é encontrado na *Tabela de Conversão de Objetos* (seção IV.3);
- *Número de Identificação da Transição de Destino*: número de identificação da transição de destino do lugar da rede em questão;
- *Duração*: tempo de duração da apresentação do OBJETO relativo a este lugar;
- *Recurso*: tipo de mídia relativa ao OBJETO, que pode ser ÁUDIO, VÍDEO, TEXTO, GRÁFICO ou IMAGEM.

Para uma transição, a descrição deve conter as seguintes informações :

- *Tipo do Componente*: identifica a descrição como sendo de uma transição;
- *Tipo da Transição*: identifica o tipo de transição que pode ser INICIAL, INTERMEDIÁRIA ou FINAL;
- *Identificação da Transição*: número de identificação da transição para a rede em questão;

- *Número de Lugares de Origem*: número de lugares de origem que necessitam conter uma ficha para disparar a transição;
- *Número de Lugares de Destino*: número de lugares que são executados ao disparo da transição;
- *Identificação dos Lugares de Destino*: contém o número de identificação de todos os lugares de destino da transição.

Lima [1] e Junqueira [20] não definiram o tipo de lugar ATRASO. Como visto anteriormente, este tipo de lugar é muito importante para a representação das sete relações temporais possíveis para a sincronização de dois intervalos de tempo. Sendo assim, foi incluído no aplicativo ***Composition*** o suporte à este tipo de lugar.

As definições das informações *Tipo do Lugar* e *Duração* na descrição de um lugar passam então a ser as seguintes:

- *Tipo do Lugar*: identifica o tipo de lugar que pode ser INICIAL, OBJETO, FINAL ou ATRASO;
- *Duração*: tempo de duração da apresentação do ATRASO ou do OBJETO relativo a este lugar.

IV.3 Tabela de Conversão de Objetos

Como exposto na seção anterior, cada lugar do tipo OBJETO possui o campo de informação *Número do Objeto* que fornece a identificação do objeto que é referenciado pelo mesmo. A Tabela de Conversão de Objetos [20] faz a associação entre o número de identificação do objeto e o arquivo que contém seus dados adicionais (seção IV.4). Esta conversão é necessária para evitar a repetição dos dados adicionais da mídia para cada arquivo de descrição do modelo OCPN.

A tabela corresponde a um arquivo ASCII denominado *ObjMM* onde, em cada linha, a primeira coluna é formada pelo número do objeto e a segunda coluna contém o nome do arquivo que contém os dados adicionais para aquele objeto. Um exemplo para esta tabela é mostrado a seguir:

Tabela IV.1: Conversão de Objetos: arquivo *ObjMM*.

1	Renata.VCR
2	renata.audio
3	bolha.text
4	mickey.image
5	Ball.graphic
6	Meta.VCR
7	bus.VCR
8	Brazil.text
9	earth.image
10	Teste.graphic
11	Flow.VCR
12	Danger.text
13	Lua.VCR
14	earthris.image
15	Clow.VCR
16	Clear.text
17	Controle.audio
18	sample.audio
19	gear.audio
20	big_house.image

IV.4 Arquivo de Dados Adicionais do Objeto

As informações contidas nos arquivos de dados adicionais de objeto dependem do tipo de recurso que pode ser áudio, vídeo, texto, imagem ou gráfico. Para o recurso ÁUDIO as informações presentes são as seguintes:

- *Tipo de Compressão* - corresponde ao algoritmo utilizado para a compressão do áudio;
- *Taxa de Amostragem* - taxa em que o áudio foi amostrado durante a sua captura;
- *Número de Bits por Amostra* - número de bits que contém cada amostra;
- *Deslocamento* - corresponde ao deslocamento em número de quadros entre o começo do vídeo gerado junto com o áudio (se houver) e o áudio;
- *Tamanho* - tamanho total do arquivo de áudio;
- *Nome* - nome do arquivo de dados do objeto.

Para o recurso VÍDEO as informações são:

- *Tipo de Compressão* - corresponde ao algoritmo utilizado para a compressão do vídeo;
- *Número de Quadros* - número total de quadros para o vídeo;
- *Número de Quadros por Segundo* - taxa de apresentação dos quadros do vídeo;
- *Largura* - largura de apresentação de cada quadro em *pixels*;
- *Altura* - altura de apresentação de cada quadro em *pixels*;
- *Nome* - nome de arquivo de dados do objeto.

Para o recurso TEXTO as informações presentes são:

- *Tamanho* - tamanho em bytes do texto;
- *Nome* - nome do arquivo de dados.

Para o recurso IMAGEM as informações são:

- *Tipo de Compressão* - corresponde ao algoritmo utilizado para a compressão da imagem;
- *Largura* - largura da imagem em *pixels*;
- *Altura* - altura da imagem em *pixels*;
- *Nome* - nome de arquivo de dados do objeto.

Por fim, para o recurso GRÁFICO as informações presentes são:

- *Ordenada* - nome dado para a ordenada;
- *Abscissa* - nome dado para a abscissa;
- *Unidade da Ordenada* - unidade em que é fornecida a escala para a ordenada;
- *Unidade da Abscissa* - unidade em que é fornecida a escala para a abscissa;
- *Escala da Ordenada* - escala da ordenada;
- *Escala da Abscissa* - escala da abscissa;
- *Formato do Gráfico* - formato em serão apresentados os dados, que pode linear, pizza, barra, entre outros;
- *Nome* - nome do arquivo de dados do objeto.

IV.5 Módulos Componentes

O diagrama da figura IV.4 mostra os principais módulos componentes do ***Composition***: Interface com o Usuário, Atribuição das Propriedades do Objeto Multimídia, Geração da OCPN e Gerência dos Arquivos do Diagrama da OCPN.

O primeiro módulo define a interface com o usuário, controlando e monitorando interativamente a construção do diagrama da OCPN. A monitoração interativa da construção do diagrama é necessária para impedir que o usuário do programa tente realizar ligações diretas entre transições, ligações diretas entre lugares e realimentação na rede.

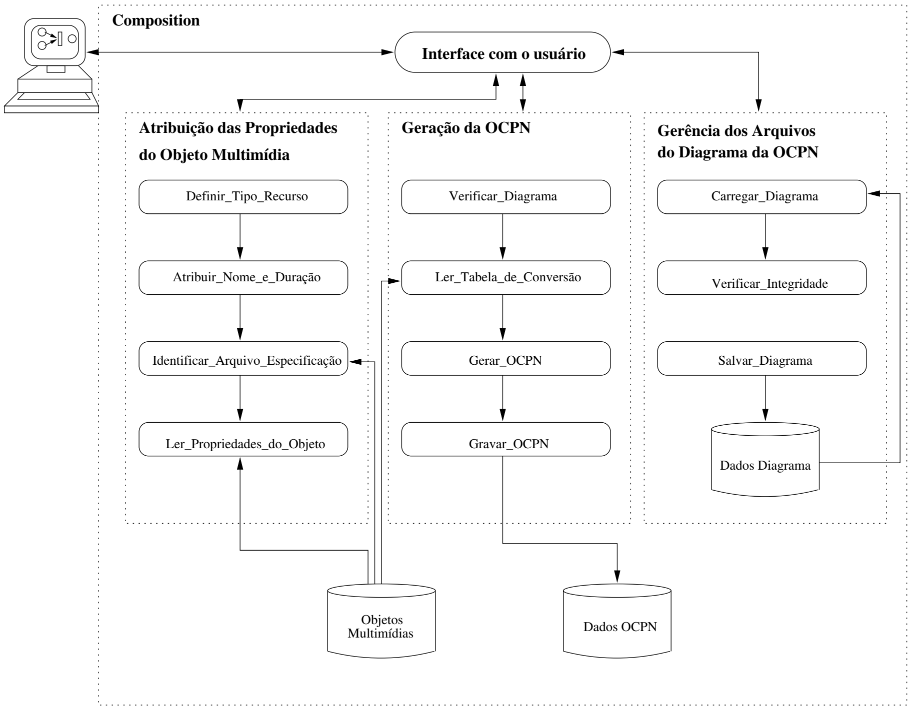

Figura IV.4: Diagrama dos principais módulos do ***Composition***.

O segundo módulo é responsável pela atribuição das propriedades dos objetos multimídias. Neste módulo é definido o tipo de lugar (inicial, final, objeto ou atraso) e, caso seja do tipo objeto, é identificado o tipo de recurso, seu nome e duração. É identificado o arquivo de especificação do objeto em questão e, finalmente, a base de dados é acessada para a leitura das propriedades do objeto.

O terceiro módulo é responsável por todo o processamento necessário à geração da especificação da OCPN. Primeiramente são realizadas as verificações necessárias na estrutura do diagrama desenhado, de forma a validar a OCPN construída e a garantir que todos os objetos possuam duração e o respectivo arquivo de especificação definidos. A validação do diagrama é realizada verificando-se: a existência dos lugares final e inicial; se todos os lugares possuem propriedades definidas; se não existem lugares sem ligações com o resto da estrutura da OCPN; se todas as transições possuem pelo menos um arco de chegada e de saída; e se todos os lugares (exceto o inicial e o final) possuem obrigatoriamente arco de chegada e de saída.

Com estas verificações realizadas, garante-se que o diagrama construído pelo autor da apresentação multimídia está montado de forma correta. Quando alguma dessas questões não são satisfeitas, todos os objetos que se enquadram no erro são automaticamente selecionados na tela e uma mensagem de erro é retornada ao usuário do programa. Após a realização de todo o processo de validação, este módulo acessa a base de dados dos objetos multimídias para ler o número de identificação de cada objeto e, em seguida, gerar a especificação da OCPN.

O quarto módulo gerencia todas as operações referentes à abertura e gravação dos arquivos que armazenam os dados necessários à remontagem do diagrama da OCPN na tela.

IV.6 Recursos

A interface do aplicativo ***Composition*** possui recursos básicos de manipulação de arquivos (*New, Open, Close, Save, Save As, Print*) acessados pelo menu *File* (Figura IV.5), possui recursos que permitem ao autor da apresentação multimídia manipular os objetos que compõem o diagrama da OCPN, como Inserir Lugar, Inserir Transição, Inserir Arco Direto, Apagar Objeto, Editar Propiedades dos Objetos, Inserir Comentário ao Diagrama da OCPN (comando *Comment* no menu *Edit* - Figura IV.6), e por fim o recurso mais importante do aplicativo que é a geração da especificação da OCPN (comando *Create* ou *Examine* no menu *Specification* - Figura IV.7).

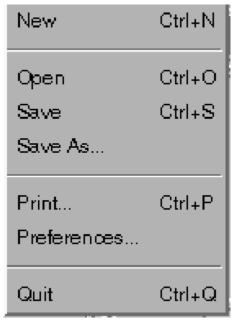

Figura IV.5: Menu *File*.

Figura IV.6: Menu *Edit*.

Figura IV.7: Menu *Specification*.

O usuário pode abrir arquivos com extensão **dgm** ou **ocpn**. Os arquivos com extensão **dgm** (diagrama) armazenam os dados necessários para a remontagem do diagrama da OCPN na tela. Já os arquivos com extensão **ocpn** contêm os dados da estrutura da OCPN em formato ASCII, isto é, contêm especificações geradas pelo ***Composition***.

Abaixo do menu principal existe uma barra de ferramentas (Figura IV.8) que permite ao usuário acessar mais rapidamente os recursos *New, Open, Save, Print, Create* e *Comment.* Além disso todas estas funções também podem ser acessadas através de teclas de atalho. Ao posicionar o apontador do *mouse* sobre algum botão da barra de ferramentas, surge uma pequena janela (*ballon*) indicando sua função.

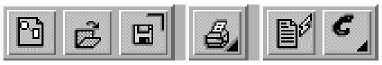

Figura IV.8: Barra de ferramentas superior.

O comando *Print* permite que o diagrama da OCPN seja impresso para arquivo no formato *postscript*. Isto é importante para permitir o uso do desenho por outros aplicativos, facilitando a conversão para outros formatos (GIF, BMP, JPG, ...).

O menu *Specification* apresenta ao usuário duas funções: geração a especificação da OCPN sem visualizá-la (comando *Create*) e geração a especificação da OCPN retornando na tela o arquivo gerado (comando *Examine*).

Com o comando *Preferences*, acessado através do menu *File*, o usuário define o diretório onde se encontra a *Tabela de Conversão de Objetos* (seção IV.3), o diretório padrão para a gravação dos arquivos de especificação da OCPN (***.ocpn**) e dos arquivos (***.dgm**), o tipo de letra utilizada nos comentários e o tipo de letra utilizado na apresentação do nome e duração do objeto multimídia. Estes dados ficam armazenados no arquivo *Composition.ini*.

IV.7 Construção de Diagramas

A barra de ferramentas lateral IV.9 facilita a inserção de lugares e transições através da utilização do botão esquerdo do *mouse*.

Para o caso de se utilizar um *mouse* de três botões, o botão central é utilizado para

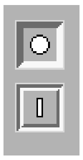

Figura IV.9: Barra de ferramentas lateral.

mover qualquer elemento da tela (transições, lugares, *labels* de lugares). A seleção do elemento é realizada automaticamente quando o apontador do *mouse* está sobre ele.

Se o *mouse* só possuir dois botões, o botão esquerdo continua com mesma função descrita anteriormente, mas para mover qualquer elemento da tela é necessário manter pressionados os botões esquerdo e direiro sendo que o botão direito deve ser pressionado primeiro.

Para realizar a ligação (arco direto) entre os lugares e transições, é necessário posicionar o apontador do *mouse* sobre o primeiro elemento (lugar ou transição) e pressionar a tecla **1**. Em seguida deve-se posicionar o apontador do *mouse* sobre o segundo elemento (transição ou lugar, de acordo com o primeiro selecionado) e pressionar a tecla **2**.

Ligações diretas entre transições, ligações diretas entre lugares e tentativa de utilizar realimentação na rede são monitoradas e indisponibilizadas interativamente à construção do diagrama da OCPN.

A figura IV.10 apresenta o diagrama de uma OCPN construído com o aplicativo.

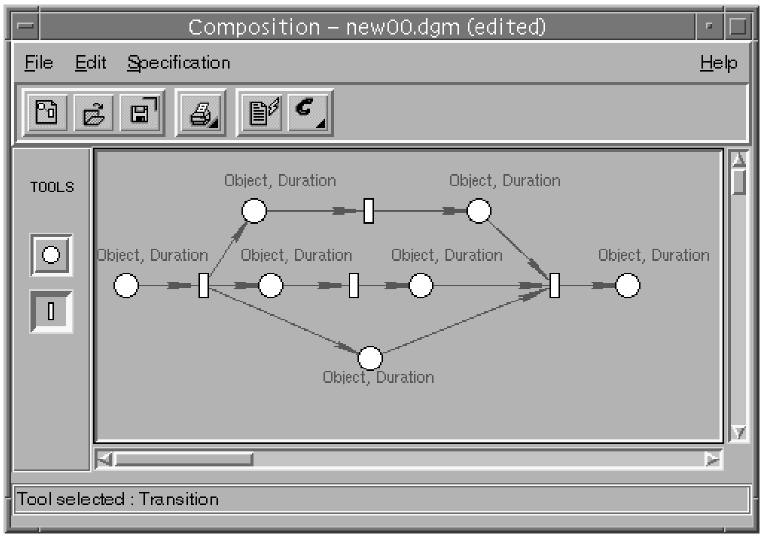

Figura IV.10: Diagrama de uma OCPN construído com o ***Composition***.

Para remover algum elemento da tela basta posicionar o apontador do *mouse* sobre o elemento em questão e pressionar a tecla DEL.

O *label* sobre os lugares da rede indica o nome do objeto e sua respectiva duração. Como não há propriedades definidas, é mostrado o *label "Object, Duration"*.

IV.8 Atribuição de Propriedades

Para acessar a janela de atribuição de propriedades, deve-se selecionar o lugar em questão e pressionar a tecla **P**. Ao surgir a janela correspondente (Figura IV.11), deve-se selecionar o tipo de lugar (Inicial, Final, Objeto ou Atraso).

A figura IV.12 apresenta como exemplo a janela de definição das propriedades de um lugar cujo recurso utilizado é o vídeo. Nesta janela, o autor da apresentação multimídia atribui o nome do objeto, sua duração e seu respectivo arquivo de especificação. Com isto, o aplicativo retorna ao autor da apresentação as propriedades do objeto que neste caso são: o arquivo de dados, o tipo de compressão, o número total de quadros, o número de quadros por segundo, a largura e a altura.

Figura IV.11: Janela inicial para a atribuição de propriedades.

Na figura IV.13 é apresentado o diagrama de uma OCPN onde todos os objetos possuem propriedades definidas. Sendo assim, não é mais mostrado o *label "Object, Duration"* sobre os lugares e sim o nome do objeto fornecido pelo autor da apresentação e sua res-

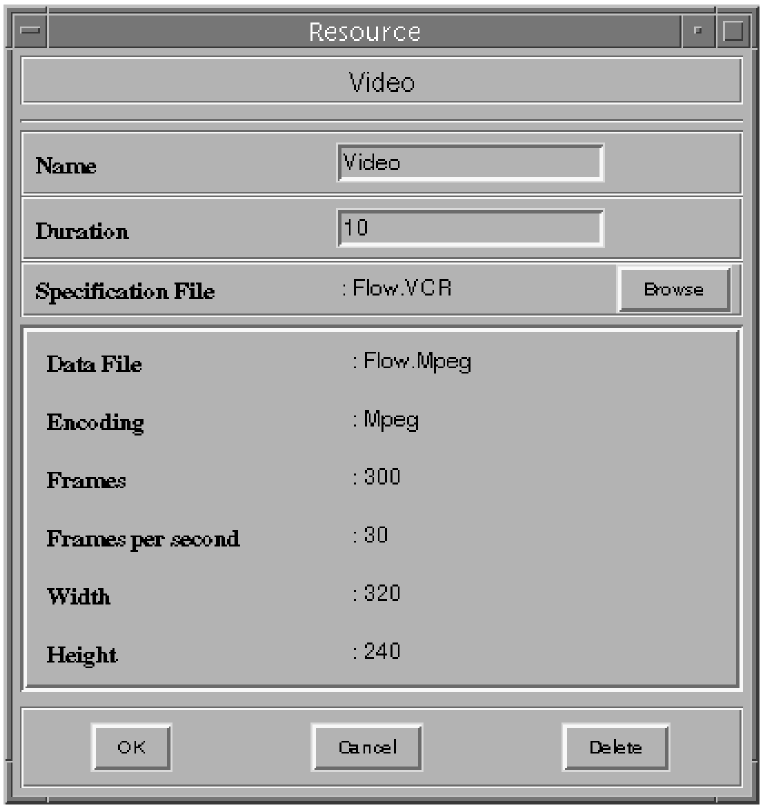

Figura IV.12: Janela de atribuição de propriedades para o recurso vídeo.

pectiva duração. Como pode ser observado, para os lugares inicial e final é mostrado o *label Initial* e *Final* respectivamente. A especificação gerada pelo aplicativo para este diagrama está listada no apêndice A.

IV.9 Formato dos Arquivos *dgm*

Os arquivos com extensão **dgm** armazenam os dados necessários à remontagem do diagrama da OCPN que foi montado pelo autor da apresentação multimídia.

Estes arquivos são gerados no formato ASCII onde a primeira linha corresponde ao campo de validação do arquivo, ou seja, nesse campo existe um código que permite ao aplicativo identificar se o referido arquivo foi ou não gerado por ele.

A segunda linha corresponde ao campo comentário. Este campo contém o comentário

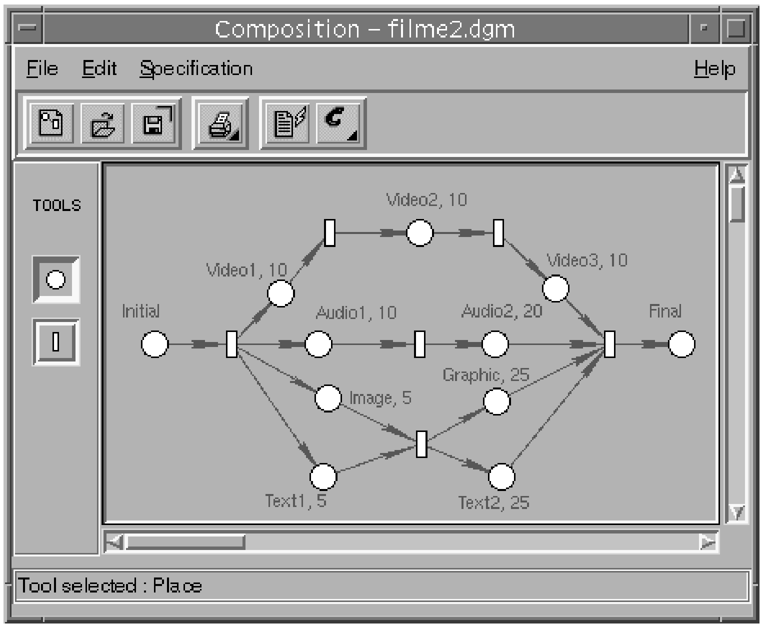

Figura IV.13: Objetos com propriedades definidas.

adicionado pelo autor da apresentação ao diagrama construído.

A partir da terceira linha são descritos os elementos que compõem o diagrama, isto é, as transições e os lugares. Para as transições, o formato da descrição é apresentado a seguir:

Número tran Arcos_Origem Arcos_Destino PosiçãoX PosiçãoY

- *Número* - Representa o número da transição. Este número é gerado pelo próprio Tcl/Tk durante a construção do diagrama da OCPN na tela;
- *tran* - Identifica a descrição como sendo de uma transição;
- *Arcos_Origem* - Contém o número de identificação dos arcos diretos que possuem origem na transição em questão. O número de identificação dos arcos diretos também é criado pelo próprio Tcl/Tk durante a construção do diagrama da OCPN na tela;
- *Arcos_Destino* - Contém o número de identificação dos arcos diretos que possuem o destino na transição em questão. O número de identificação dos arcos diretos é criado pelo próprio Tcl/Tk durante a construção do diagrama da OCPN na tela;

- *PosiçãoX* - Posição X da transição na tela do aplicativo;
- *PosiçãoY* - Posição Y da transição na tela do aplicativo.

Para os lugares, o formato da descrição é:

Número lugar Arco_Origem Arco_Destino PosiçãoX PosiçãoY PosiçãoX_Label PosiçãoY_Label Propriedades

- *Número* - Representa o número do lugar que é gerado pelo Tcl/Tk na fase de construção do diagrama da OCPN na tela;
- *lugar* - Identifica a descrição como sendo de um lugar;
- *Arco_Origem* - Contém o número de identificação do arco direto que possui origem no lugar em questão. O número de identificação do arco direto também é criado pelo próprio Tcl/Tk durante a construção do diagrama da OCPN na tela;
- *Arco_Destino* - Contém o número de identificação do arco direto que possui o destino no lugar em questão. O número de identificação do arco direto é criado pelo Tcl/Tk durante a construção do diagrama da OCPN na tela;
- *PosiçãoX* - Posição X do lugar na tela do aplicativo;
- *PosiçãoY* - Posição Y do lugar na tela do aplicativo;
- *PosiçãoX_Label* - Posição X do *label* do lugar na tela do aplicativo;
- *PosiçãoY_Label* - Posição Y do *label* do lugar na tela do aplicativo;
- *Propriedades* - Propriedades dos lugar em questão.

A seguir é apresentado o arquivo *filme2.dgm* que contém os dados referentes ao diagrama mostrado na figura IV.13.

Composition_091174xb3_

Diagrama do filme 2.

17 lugar 39 33 194 43 174 23 {VIDEO Lua.VCR Video2 10 Jpeg 200 30 320 240 Lua.Jpeg}

1 tran 29 {30 32 35 36} 310 111

2 lugar 30 42 241 146 208 130 {GRAFICO Ball.graphic Graphic 25 {Funcão Densidade de Probabilidade} Atraso {} seconds 0.05 1 linear Ball.grh}

19 lugar 40 {} 32 111 12 91 INICIAL

21 tran {41 42} {34 43} 195 172

22 lugar 43 46 138 144 151 144 {IMAGEM earthris.image Image 5 GIF 320 200 earthris.gif}

4 lugar 31 45 132 111 131 92 {AUDIO sample.audio Audio1 10 COMPRESSED_ULAW {8000 Hz} 8 0 {876338 bytes} sample.au}

24 lugar 44 48 109 80 64 66 {VIDEO Clow.VCR Video1 10 Jpeg 200 30 320 240 Clow.Jpeg}

6 lugar 32 37 240 111 220 91 {AUDIO gear.audio Audio2 20 COMPRESSED_ULAW {8000 Hz} 8 0 {3008 bytes} gear.au}

26 lugar {} 29 354 111 334 91 FINAL

8 tran 33 44 139 43

9 lugar 34 47 135 192 100 207 {TEXTO Danger.text Text1 5 4769 Danger.txt}

28 tran {45 46 47 48} 40 79 111

11 lugar 35 38 277 77 272 60 {VIDEO bus.VCR Video3 10 Jpeg 200 30 320 240 bus.Jpeg}

13 lugar 36 41 244 192 218 208 {TEXTO Brazil.text Text2 25 2904 Brazil.txt}

15 tran 37 31 194 111

16 tran 38 39 242 43

Os números de identificação atribuídos pelo Tcl/Tk aos elementos componentes do diagrama da OCPN podem ser quaisquer desde que não sejam repetidos. Isto não influencia na descrição da OCPN gerada pelo aplicativo, entretanto, um arquivo já existente que seja reaberto e salvo imediatamente pode ser diferente do arquivo anterior devido a este descontrole por parte do programa com relação aos números de identificação gerados. Tudo isto é transparente ao usuário, uma vez que para ele, os diagramas sempre serão idênticos.

Capítulo V

Considerações Finais

O ***Composition*** é parte integrante dos trabalhos de pesquisa e estímulos do Grupo de Teleinformática e Automação na área de Sincronização Multimídia. Os trabalhos desenvolvidos nesta área visam a implementação de um Ambiente Integrado de Apresentações Multimídias. Suporte à captura, manipulação, composição, integração, armazenamento, comunicação e apresentação sincronizada das mídias são etapas essenciais à implementação deste ambiente integrado.

Alguns processos relativos aos mecanismos de sincronização foram automatizados, como o serviço de geração das XOCPNs de transmissão e recepção. Dando continuidade às etapas de implementação acima citadas, o ***Composition*** surgiu como uma ferramenta gráfica que permite a especificação formal da sincronização entre os objetos multimídias e a modelagem completa das apresentações que se deseja elaborar.

O modelo OCPN, sobre o qual o aplicativo ***Composition*** está fundamentado, demonstrou-se suficientemente adequado para a especificação dos requisitos temporais dos objetos multimídias à nível de aplicação. O seu enfoque está na sincronização intermídia fornecendo um método geral para especificar a sincronização e realizar a composição temporal entre os objetos multimídias. A extensão para este modelo, XOCPN, divide os objetos multimídias em sub-objetos para melhor coordenar e manter as relações temporais contínuas entre as diversas mídias, mas para tal tarefa é necessária a utilização dos dados obtidos a partir da OCPN. Isto evidencia uma das importâncias do aplicativo

Composition para o ambiente integrado de apresentações multimídias.

O conjunto de ferramentas de programação Tcl/Tk desempenhou um papel importante na implementação do aplicativo, facilitando a implementação de todo o processo de validação do diagrama da OCPN construído pelo autor da apresentação multimídia.

É importante ressaltar que o Tcl/Tk são linguagens interpretadas, sendo assim, foi necessário avaliar a necessidade de utilização de procedimentos escritos em C ou em Tcl. Como para a implementação de todas as funcionalidades do aplicativo não houve a necessidade de se utilizar cálculos complexos que influenciassem diretamente no desempenho do sistema, embora houvesse funções complexas de se implementar, não foi necessário agregar procedimentos C ao núcleo Tcl.

Saber quando usar C ou Tcl/Tk é uma etapa primordial do desenvolvimento de qualquer aplicativo que utiliza estas linguagens. Com isto, as vantagens de se utilizar o Tcl/Tk tornam-se evidentes. O tempo economizado e a pouca quantidade de código necessária à implementação da interface gráfica, cerca de 320 linhas, e à implementação dos recursos oferecidos ao autor das apresentações multimídias, cerca de 1700 linhas, foram ganhos claramente observados.

É comum que um programa escrito para uma máquina seja transportado para outra que possua o processador ou o sistema operacional diferente, ou ambos. Esse processo é chamado portabilidade e pode ser muito fácil ou extremamente difícil de ser obtido, dependendo de como o programa foi originalmente escrito. Um programa que pode ser facilmente transportado é chamado portável. Um programa não é portável quando contém numerosas dependências da máquina - fragmentos de código que funcionem somente com um sistema operacional ou processador específico. Foi possível desenvolver o aplicativo ***Composition*** de forma independente do sistema operacional e plataforma utilizados. Isto se deve a duas características:

- as ferramentas de programação Tcl/Tk são linguagens interpretadas que, dependendo de como o programa desenvolvido foi originalmente escrito, permite sua utilização em qualquer máquina que possua o interpretador Tcl/Tk instalado;

- a habilidade utilizada para desenvolver o aplicativo com fragmentos de código comuns aos vários sistemas operacionais e plataformas.

Com estas características amplamente exploradas, o ***Composition*** funcionou perfeitamente em computadores PC com Linux ou FreeBSD e em estações de trabalho Sun Sparc 20 com Solaris 2.4 ou Solaris 2.5 instalado.

O ***Composition*** oferece um ambiente gráfico e simples para que autores de apresentações multimídias realizem de forma ágil e segura a composição dos objetos multimídias que farão parte da apresentação que se deseja criar, sendo desta forma de vital importância para o ambiente integrado que está sendo desenvolvido.

A busca por novas aplicações, técnicas e métodos de sincronização multimídia com base no grande avanço da tecnologia digital e da microeletrônica é importante para a proposição de novas soluções e aperfeiçoamentos que só contribuirão para o desenvolvimento dos Sistemas Multimídias. Sendo assim, como proposta para trabalhos futuros, poderiam ainda ser implementados o suporte à criação de hierarquias através da construção de subredes no diagrama da OCPN, como abordado na seção III.5 (Figura III.6), e a implementação do mapeamento da rede construída pelo autor da apresentação multimídia em outros métodos de especificação, se fazendo valer sempre das vantagens que estes métodos irão proporcionar ao usuário do programa e ao Sistema Multimídia como um todo.

O Grupo de Teleinformática a Automação dispõe agora de um novo patamar a partir do qual novas prioridades e demandas com relação ao ambiente integrado poderão ser identificadas. Apresentações como vídeo-clips, aulas experimentais em laboratórios, projeção de imagens com áudio explicativo e textos com versão em diferentes idiomas são objetivos que estão na iminência de serem alcançados.

Referências Bibliográficas

[1] R. M. de Lima, "Um Serviço de Sincronização Multimídia de Dados Pré-Orquestrados", Tese de Mestrado, PEE-COPPE/UFRJ, junho 1996.

[2] R. Steinmetz, "Synchronization Properties in Multimedia Systems", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 401–412, abril 1990.

[3] G. Blakowski e R. Steinmetz, "A Media Synchronization Survey: Reference Model, Specification, and Case Studies", *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 1, pp. 5–35, janeiro 1996.

[4] R. M. de Lima, F. P. Junqueira, P. A. S. Gonçalves e O. C. M. B. Duarte, "O Ambiente Integrado SAMM de Suporte à Sincronização de Dados Multimídias Pré-Orquestrados", Relatório Técnico, GTA-COPPE/UFRJ, 1998.

[5] C. Nicolaou, "An Architecture for Real-Time Multimedia Communication Systems", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 391–400, abril 1990.

[6] J. Escobar, C. Partridge e D. Deutsch, "Flow Synchronization Protocol", *IEEE/ACM Transactions on Network*, vol. 2, no. 2, pp. 111–121, abril 1994.

[7] F. E. Horn e J. B. Stefani, "On Programming and Supporting Multimedia Object Synchronization", *The Computer Journal*, vol. 36, no. 1, pp. 4–18, 1993.

[8] D. P. Anderson e G. Homsy, "A Continuos Media I/O Server and Its Synchronization Mechanism", *IEEE Computer*, pp. 51–57, 1991.

[9] S. Ramanathan e P. V. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems", *The Computer Journal*, vol. 36, no. 1, pp. 19–31, fevereiro 1993.

[10] D. Shepherd, D. Hutchinson, F. Garcia e G.Coulson, "Protocol Support for Distributed Multimedia Applications", *Comput. Commun.*, vol. 15, no. 6, pp. 359–366, julho 1992.

[11] T. D. C. L. e A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 413–426, abril 1990.

[12] L. F. R. C. Carmo, J. P. Courtiat e P. Saqui-Sannes, "Towards Multimedia Communication Services", in *Proc. Of the Fourth Workshop on Future Trends of Distributed Computing Systems*, Lisboa, Portugal, setembro 1993.

[13] N. U. Qazi, M. Woo e A. Ghafoor, "A Synchronization and Communication Model for Distributed Multimedia Objects", in *Proc. ACM Multimedia '93*, Anaheim, CA, pp. 147–156, agosto 1993.

[14] R. Steinmetz e K. Nahrstedt, *Multimedia: Computing, Communications & Applications.* Prentice-Hall, 1995.

[15] R. M. de Lima e O. C. M. B. Duarte, "Implementação de um Serviço de Sincronização Multimídia para Comunicação de Dados Pré-Orquestrados", in *XV Simpósio Brasileiro de Redes de Computadores SBRC'97*, São Carlos, SP, pp. 19–35, maio 1997.

[16] M. Woo, N. U. Qazi e A. Ghafoor, "A Synchronization framework for Communication of Pre-Orchestrated Multimedia Information", *IEEE Network*, vol. 8, no. 1, pp. 52–61, janeiro 1994.

[17] R. C. Teixeira, "Um Serviço de Sincronização Interfluxo para Comunição Multimídia", Projeto Final de Graduação, DCC/IM/UFRJ, fevereiro 1997.

[18] M. G. Rubinstein, "Um Serviço de Sincronização Contínua para Aplicações Multimídias", Tese de Mestrado, COPPE/UFRJ, novembro 1996.

[19] M. G. Rubinstein e O. C. M. B. Duarte, "Implementação de um Serviço de Sincronização Baseado em Relações Temporais Implícitas", in *II Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Fortaleza, CE, pp. 282–293, novembro 1997.

[20] F. P. Junqueira, "Geração Automática de XOCPN para um Serviço de Sincronização", Projeto Final de Graduação, DEL/EE/UFRJ, fevereiro 1997.

[21] B. Welch, *Pratical Programming in Tcl and Tk.* Addison-Wesley Publishing Company, abril 1995.

[22] J. K. Ousterhout, *Tcl and the Tk Toolkit.* Addison-Wesley Publishing Company, 1994.

[23] T. D. C. L. e A. Ghafoor, "Network considerations for distributed multimedia object composition and communication", *IEEE Network*, vol. 4, no. 6, pp. 32–49, novembro 1990.

[24] T. D. C. Little e A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks", *IEEE Computer*, pp. 42–50, outubro 1991.

[25] T. D. C. Little e A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services", *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 9, pp. 1368–1382, dezembro 1991.

[26] T. D. C. Little e A. Ghafoor, "Scheduling of Bandwidth-Constrained Multimedia Traffic", *Comp. Commun*, vol. 15, no. 6, pp. 381–387, julho 1992.

[27] R. M. de Lima e O. C. M. B. Duarte, "Comunicação Sincronizada de Dados Pré-Orquestrados", in *XXIII SEMISH*, Recife, PE, pp. 285–296, agosto 1996.

[28] J. L. Szwarcfiter, *Grafos e Algoritmos Computacionais.* Editora Campus, 1988.

Apêndice A

Listagem

Neste apêndice é apresentada a especificação da OCPN gerada pelo ***Composition*** para o diagrama da figura IV.13.

```
#
# Este arquivo foi gerado pelo aplicativo Composition.
#

Lugar
Tipo : INICIAL
Identificacao_Lugar : 6
Numero_Objeto : 0
Identificacao_Tran_Dest : 22
Duracao :
Recurso :

Lugar
Tipo : OBJETO
Identificacao_Lugar : 13
```

```
Numero_Objeto : 15

Identificacao_Tran_Dest : 19

Duracao : 10

Recurso : VIDEO


Lugar

Tipo : OBJETO

Identificacao_Lugar : 4

Numero_Objeto : 5

Identificacao_Tran_Dest : 3

Duracao : 25

Recurso : GRAFICO


Lugar

Tipo : OBJETO

Identificacao_Lugar : 23

Numero_Objeto : 7

Identificacao_Tran_Dest : 3

Duracao : 10

Recurso : VIDEO


Lugar

Tipo : OBJETO

Identificacao_Lugar : 9

Numero_Objeto : 14

Identificacao_Tran_Dest : 8

Duracao : 5

Recurso : IMAGEM


Lugar

Tipo : OBJETO
```

```
Identificacao_Lugar : 1
Numero_Objeto : 13
Identificacao_Tran_Dest : 28
Duracao : 10
Recurso : VIDEO

Lugar
Tipo : OBJETO
Identificacao_Lugar : 20
Numero_Objeto : 12
Identificacao_Tran_Dest : 8
Duracao : 5
Recurso : TEXTO

Lugar
Tipo : OBJETO
Identificacao_Lugar : 15
Numero_Objeto : 19
Identificacao_Tran_Dest : 3
Duracao : 20
Recurso : AUDIO

Lugar
Tipo : OBJETO
Identificacao_Lugar : 11
Numero_Objeto : 18
Identificacao_Tran_Dest : 27
Duracao : 10
Recurso : AUDIO

Lugar
```

```
Tipo : OBJETO
Identificacao_Lugar : 25
Numero_Objeto : 8
Identificacao_Tran_Dest : 3
Duracao : 25
Recurso : TEXTO

Lugar
Tipo : FINAL
Identificacao_Lugar : 17
Numero_Objeto : 0
Identificacao_Tran_Dest :
Duracao :
Recurso :

Transicao
Tipo : INICIAL
Identificacao_Transicao : 22
Numero_Lugares_Origem : 1
Numero_Lugares_Destino : 4
Identificacao_Lugares_Destino : 9 11 13 20

Transicao
Tipo : INTERMEDIARIA
Identificacao_Transicao : 19
Numero_Lugares_Origem : 1
Numero_Lugares_Destino : 1
Identificacao_Lugares_Destino : 1

Transicao
Tipo : INTERMEDIARIA
```

```
Identificacao_Transicao : 8
Numero_Lugares_Origem : 2
Numero_Lugares_Destino : 2
Identificacao_Lugares_Destino : 4 25

Transicao
Tipo : INTERMEDIARIA
Identificacao_Transicao : 27
Numero_Lugares_Origem : 1
Numero_Lugares_Destino : 1
Identificacao_Lugares_Destino : 15

Transicao
Tipo : INTERMEDIARIA
Identificacao_Transicao : 28
Numero_Lugares_Origem : 1
Numero_Lugares_Destino : 1
Identificacao_Lugares_Destino : 23

Transicao
Tipo : FINAL
Identificacao_Transicao : 3
Numero_Lugares_Origem : 4
Numero_Lugares_Destino : 1
Identificacao_Lugares_Destino : 17
```