

*Especificação de Aplicações Hiperfídia
Baseada em Statecharts*

Fabiano Borges Paulo

Orientador: Prof. Dr. Paulo Cesar Masiero

Dissertação apresentada ao Instituto de Ciências Matemáticas de São Carlos - USP como parte dos requisitos para a obtenção do título de mestre na Área de Ciências de Computação e Matemática Computacional.

USP - São Carlos
- 1997 -

Aos meus pais, Laurentino e Therezinha

À Sandra

Agradecimentos

Ao Senhor Jesus, pela vida, pelas oportunidades, pelas pessoas que tem colocado em minha vida e pelas bênçãos e graças que nunca cessaram.

Aos meus pais, Laurentino e Therezinha, pelo amor, pela compreensão, pelo apoio incondicional às minhas decisões, pela educação e formação que me proporcionaram, muitas vezes com sacrifício, e pelo exemplo de vida que são.

À Sandra Kawamoto, pelo amor, pelo companheirismo, pela ajuda e pela paciência.

Ao professor Paulo Masiero, por ter-me confiado este trabalho, pela orientação, tanto acadêmica quanto pessoal, pelas lições em cada conversa, por ter-me cedido sua sala e por toda ajuda que me tem dado.

Ao professor José Carlos Maldonado, pelo incentivo constante.

À professora e amiga Rosângela Penteado pelo incentivo, pelo apoio, pelos conselhos, pela ajuda desde a graduação, pela abertura desta oportunidade e pela convivência inesquecível.

Ao Tchelo, pela parceria no projeto da HySCharts e pelas várias e valorosas colaborações.

Aos amigos e colegas, entre eles os mais próximos: Dirceu Penteado, Sandra Fabbri, Simone, Paulo, Luciano, Elisa, Delamaro, Djalma, Paulo Misaka, Gawa, Mauro, "Trolha", Caldas, Marisa, Júlio, Victor, Marcos Quinaia, Donizete Bruzarosco, Doniseti, Renato, Cris, Chandler, Jaque, Rejane, Márcia, Daniel, Rudinei e todos os demais pela convivência, pelas trocas de idéias e pela solidariedade.

A todos que torceram a favor, entre eles os padrinhos Gilberto e Zeneide e todos em São Caetano, os tios Paulo, Dirce e todos em Sorocaba, tia "Nega" e filhos, os sogros Luiz e Cecília e os cunhados Luciane e Reginaldo.

Aos professores do ICMSC-USP, pelas disciplinas da pós-graduação e pelo trabalho administrativo que realizam para que tudo funcione da melhor forma. Agradecimento especial às professoras Maria Cristina de Oliveira, parceira nos artigos, e Renata Fortes pelas sugestões e colaborações no decorrer deste trabalho.

A todos os funcionários, principalmente os do ICMSC-USP, que direta ou indiretamente, colaboraram para a realização deste trabalho. Um agradecimento especial à Beth e à Laura, da seção de pós-graduação, pelo trabalho irrepreensível, ao Paulinho, da seção de alunos, pela amizade, à Sandrinha, da seção de bolsas, pela assistência e ao Rogério, do LDCC, pela cooperação e pela paciência.

Ao Rogério Hiraoka, aluno de iniciação científica, pela ajuda com a verificação dos aspectos práticos do modelo e pelo trabalho que vem realizando, garantindo a continuidade deste projeto.

Ao CNPq e a Fapesp pelo suporte financeiro.

Resumo

Uma classificação de requisitos de aplicações hipermídia é apresentada e alguns dos mais importantes modelos de especificação desse tipo de aplicação são revisados com base nessa classificação. Uma análise do HMBS (“Hypertext Model Based on Statecharts”) frente à classificação de requisitos apresentada é também realizada. Uma extensão do HMBS para a especificação de aplicações hipermídia, denominada XHMBS (“Extended Hyperdocument Model Based on Statecharts”), é proposta e, no final, também comparada com essa mesma classificação de requisitos. Esse modelo tem como base uma extensão de statecharts proposta nesta dissertação, denominada *hypercharts*, na qual são definidos mecanismos de especificação temporal e de sincronização, como *história temporal*, *transições temporais* e *transições M:N sincronizadas*, além de mecanismos de abstração, caracterizados por *estados parametrizados* e *abstração de transições*. Os *hypercharts* constituem a técnica de especificação subjacente ao XHMBS, permitindo que requisitos temporais e de sincronização sejam satisfeitos. Além disso, novos recursos de especificação que fazem do XHMBS um modelo mais completo e, ao mesmo tempo, mais flexível do que o HMBS para especificar requisitos relativos à apresentação da informação, são apresentados. Um exemplo utilizado na literatura como base para comparações é especificado em XHMBS e os resultados comparados com relação à quantidade de símbolos usados na especificação (complexidade) e à capacidade de especificar os requisitos da aplicação.

Abstract

A classification of hypermedia applications requirements is presented and, based on this classification, some of the most important specification models for this kind of application are reviewed. An analysis of the HMBS (Hypertext Model Based on Statecharts) with respect to the presented requirements classification is also presented. An extension of the HMBS for hypermedia applications modeling called XHMBS (Extended Hyperdocument Model Based on Statecharts) is proposed and, at the end, it is compared to this same requirements classification. This model is based on a statecharts extension called *hypercharts*, proposed in this dissertation, in which temporal and synchronization specification mechanisms are defined, like *timed history*, *timed transitions* and *M:N synchronized transitions*. Abstraction mechanisms, namely, *parameterized states* and *transition abstractions* are defined too. *Hypercharts* are the XHMBS underlying design technique, allowing the satisfaction of temporal and synchronization requirements. Moreover, brand new specification resources that make XHMBS a more complete and flexible model than HMBS for specifying information presentation requirements are presented. An example used in the literature as a benchmark is specified in XHMBS and the results are compared with other specification methods. The basis for this comparison is the number of symbols used in the specification (checking complexity) and power to specify the application requirements.

Índice

Capítulo 1 - Introdução	1
1.1 Situação do problema	1
1.2 Objetivos	3
1.3 Organização	4
Capítulo 2 - Classificação dos requisitos de aplicações hipermedia	5
2.1 Considerações iniciais	5
2.2 Requisitos relativos à obtenção e organização da informação	5
2.2.1 Modelagem do domínio da aplicação	5
2.2.2 Derivação de elementos da aplicação	5
2.2.3 Estruturação da informação	6
2.2.4 Separação entre estrutura e conteúdo da informação	6
2.2.5 Várias visões da mesma informação	6
2.2.6 Versões "sob medida" do mesmo hipermento	7
2.2.7 Especificação em vários níveis de abstração	7
2.2.8 Especificação em vários níveis de granularidade	7
2.3 Requisitos relativos à sincronização da apresentação	7
2.3.1 Requisitos de sincronização relativos à cadeia de dados	8
2.3.1.1 Local de origem	8
2.3.1.2 Local de destino	8
2.3.1.3 Intervalos de apresentação	9
2.3.1.4 Tipos de fonte	9
2.3.1.5 Dependência temporal	9
2.3.2 Requisitos relativos à comunicação	10
2.3.2.1 Atraso	10
2.3.2.2 "Jitter"	10

2.3.2.3 Taxas de “throughput”	10
2.3.2.4 Sensibilidade a erros	10
2.4 Requisitos relativos à interação entre usuário e computador	11
2.5 Requisitos relativos à apresentação da informação	11
2.5.1 Coordenação espacial	11
2.5.2 Controle dos atributos globais da apresentação	11
2.5.3 Tipo de mídia da cadeia de dados	12
2.6 Requisitos relativos à navegação do hiperdocumento	12
2.6.1 Noção de visita a um nó	12
2.6.2 Estruturas alternativas de acesso à informação	12
2.6.3 Ligações temporais	12
2.6.4 Ligações dinâmicas	12
2.6.5 Ligações dentro de componentes	13
2.6.6 Contextos de informação	13
2.7 Considerações finais	13
Capítulo 3 - Modelos para especificação de aplicações de multimídia, hipertexto e hiperídia	15
3.1 Considerações iniciais	15
3.2 Modelos baseados em Redes de Petri	17
3.2.1 “Timed Petri Nets”	18
3.2.2 “Time Petri Nets”	18
3.2.3 “Arc Time Petri Nets”	19
3.2.4 O modelo OCPN	20
3.2.5 O modelo Trellis	22
3.2.6 O modelo XOCPN	23
3.2.7 O modelo DTPN	24
3.2.8 O modelo TSPN	26
3.2.9 O modelo TSPN _{UI}	27
3.2.10 O modelo MHPN	29

3.2.11 O modelo HTSPN	30
3.3 Modelos baseados em outras técnicas	32
3.3.1 Os modelos HDM e RMDM	32
3.3.2 O modelo de referência Dexter para hipertexto	33
3.3.3 O "Amsterdam Hypermedia Model"	34
3.4 Statecharts e hipermídia	35
3.4.1 Visão geral de statecharts	36
3.4.2 Statecharts temporais	40
3.4.3 O modelo HMBS	41
3.4.4 O modelo HMBS frente aos requisitos de aplicações hipermídia	43
3.4.4.1 Requisitos relativos à obtenção e organização da informação	43
3.4.4.2 Requisitos relativos à sincronização da apresentação	45
3.4.4.3 Requisitos relativos à interação entre usuário e computador	46
3.4.4.4 Requisitos relativos à apresentação da informação	46
3.4.4.5 Requisitos relativos à navegação do hiperdocumento	46
3.4.4.6 Resumo da análise do HMBS	48
3.5 Considerações finais	50
Capítulo 4 - Hypercharts	51
4.1 Especificação em alto nível baseada em statecharts	51
4.2 A base temporal dos <i>hypercharts</i>	51
4.2.1 História temporal	53
4.2.2 Verificação temporal	54
4.3 Transições temporais	55
4.4 Mecanismos de sincronização	62
4.4.1 Sincronização do tipo <i>Strong-or</i>	66
4.4.2 Sincronização do tipo <i>Weak-and</i>	68
4.4.3 Sincronização do tipo <i>Master</i>	69
4.4.4 Sincronização do tipo <i>Or</i>	71

4.4.5 Sincronização do tipo <i>And</i>	73
4.4.6 Sincronização do tipo <i>Or-Master</i>	74
4.4.7 Sincronização do tipo <i>Strong-Master</i>	76
4.4.8 Sincronização do tipo <i>Weak-Master</i>	77
4.4.9 Sincronização do tipo <i>And-Master</i>	78
4.5 Parametrização de estados e abstração de transições	79
4.5.1 Estados parametrizados	80
4.5.2 Abstração de transições	83
4.5.3 Estados parametrizados com abstração de transições	85
4.5.4 Parametrização com notação temporal, mecanismos de sincronização e abstração de transições	87
4.6 Considerações finais	90
Capítulo 5 - Especificação do modelo XHMBS	91
5.1 Um modelo para documentos de hiperídia	91
5.2 O hiperdocumentos XHMBS	91
5.3 As características do XHMBS	94
5.3.1 Melhor definição de páginas	94
5.3.2 Várias páginas associadas ao mesmo estado	94
5.3.3 Âncoras como unidades independentes da estrutura	94
5.3.4 Pchannels e Cchannels	95
5.3.4.1 Características comuns aos pchannels	96
5.3.4.2 Características particulares de cada pchannel	98
5.3.4.3 Cchannels	100
5.3.5 As relações de visibilidade	101
5.3.6 A semântica de navegação do XHMBS	103
5.4 A notação do modelo XHMBS	104
5.5 Interações com o usuário no modelo XHMBS	108
5.6 Um exemplo de especificação com XHMBS	112
5.6.1 Descrição do hiperdocumento	112

5.6.2 A apresentação multimídia	112
5.6.3 Comparação de resultados	121
5.6.4 O modelo XHMBS e os requisitos de aplicações hipermídia	123
5.6.4.1 Requisitos relativos à obtenção e organização da informação	123
5.6.4.2 Requisitos relativos à sincronização da apresentação	125
5.6.4.3 Requisitos relativos à interação entre usuário e computador	127
5.6.4.4 Requisitos relativos à apresentação da informação	128
5.7 Implementação de especificações XHMBS	132
5.7.1 Suporte traduzido ao XHMBS	133
5.7.2 Suporte interpretado ao XHMBS	134
5.8 Considerações finais	135
Capítulo 6 - Conclusões	136
6.1 Considerações finais	136
6.2 Contribuições deste trabalho	137
6.3 Sugestões de novas pesquisas	138
Bibliografia	140
Apêndice A	143
Apêndice B	148

Capítulo 1

Introdução

1.1 Situação do Problema

A base dos relacionamentos entre indivíduos é a comunicação, ou seja, a troca de informações. São várias as razões que acarretam a troca de informações, assim como são várias as formas através das quais elas ocorrem. Hoje em dia, com o advento da chamada info-era, o aumento da importância da informação e a disponibilidade de tecnologia estimulam as formas automatizadas de comunicação. As novas tecnologias em redes de computadores e máquinas de alto desempenho disponibilizam recursos de comunicação que constituem uma constante evolução na forma de interação entre usuários de sistemas computadorizados.

Os cinco sentidos do ser humano funcionam como canais pelos quais as informações podem fluir, de modo que as formas de comunicação sempre exploram alguns desses sentidos. A visão é, para a maioria das pessoas, o sentido mais utilizado em suas comunicações, seguido possivelmente da audição. Isso explica o fato de que a maior parte das trocas de informações até hoje é feita oralmente ou através de documentos escritos e imagens.

A mente humana organiza informações de maneira associativa, ou seja, as porções de informação que adquirimos através de alguns dos nossos sentidos são armazenadas e ligadas a outras porções já existentes em nossa mente, formando uma rede de informações. O fato de as pessoas se recordarem de algumas informações (imagens, odores, sons) a partir de uma informação inicial pode ser representado pelo percurso das ligações entre essas porções de informação armazenadas nessa rede mental de conhecimento. Dessa forma, os processos de escrita e de leitura para, respectivamente, expressar e adquirir conhecimentos podem ser considerados formas artificiais de comunicação. O que ocorre com a escrita é que a informação, organizada na mente em forma de rede, sofre uma linearização para se tornar um texto [Gin95]. Esse texto, por sua vez, é lido, passando por uma deslinearização, e a sua interpretação é armazenada na mente do leitor na forma de uma rede não necessariamente igual àquela residente na mente do autor. Pode-se dizer que, quanto maior o texto, mais diferenças existirão entre as redes de conhecimento do autor e do leitor, visto que a quantidade de nós aumenta a complexidade das redes. Assim, uma melhoria na comunicação textual poderia ser obtida se os textos fossem organizados em porções menores e interligados através de algum mecanismo, de forma que fosse possível recuperar a informação na forma em que ela está organizada na mente, amenizando a artificialidade do texto linear convencional.

Nos anos quarenta, um cientista americano chamado Vannevar Bush especulou sobre a criação, no futuro, de um dispositivo mecânico chamado "memex", capaz de manipular microfílmes interligados através de células foto-sensíveis [Bus45]. O leitor poderia visualizar porções de textos gravadas nos microfílmes e, a partir do texto apresentado, selecionar outras porções que seriam recuperadas pelo mecanismo através da ativação de uma combinação de foto-células. Apesar dessa idéia sugerir somente uma engenhoca, ela constituiu-se para os contemporâneos da info-era em uma nova concepção de organização e recuperação de informações: o hipertexto, uma forma de organização não linear da informação associada com a possibilidade de interação com o leitor.

A idéia de Bush surgiu durante a primeira era da computação, quando o custo da tecnologia era muito alto e poucas organizações tinham acesso aos computadores e seus benefícios. Com a difusão da computação, houve um aumento da demanda por softwares de suporte a negócios, os chamados sistemas de informação. As organizações passaram a incorporar cada vez mais esse tipo de software para suportar as suas comunicações, de forma que hoje em dia, os sistemas de informação estão por toda parte. Com a tecnologia mais avançada, a idéia de Bush pôde ser implementada por software, permitindo que os sistemas de hipertexto tomassem lugar definitivo na computação atual. Hoje em dia é comum encontrar-se enciclopédias, livros, dicionários e vários outros produtos, consagrados em suas formas textuais, em papel, apresentados em forma de hipertextos em CD-ROMs. Mais importante ainda vem sendo o impacto causado pela porção interativa da Internet, a WWW ("World Wide Web"), disponibilizando milhões de páginas eletrônicas baseadas em hipertexto por todo o planeta.

A idéia de não linearidade sugerida por Bush não mudou até hoje e é a grande característica dos hipertextos. O que mudou foi a forma de interação com o usuário. Atualmente, os dispositivos de hardware para interação como "mouse", "touch-screens" e dispositivos de realidade virtual; as tecnologias de armazenamento e comunicação de dados e as interfaces gráficas permitem uma forma muito mais agradável, ágil e abrangente de interação entre usuário e sistema.

Em termos computacionais, pode-se dizer, grosseiramente, que um sistema de hipertexto possui dois componentes: a hiperbase e o sistema que a gerencia. A hiperbase é a base de dados de um hipertexto. Nela são armazenadas as porções de informação pura que serão apresentadas aos usuários e meta informações, que determinam as ligações entre as porções de informação. O sistema gerenciador da hiperbase deve suportar operações de criação e recuperação dos dados, ou seja, autoria e navegação de hiperdocumentos. Um hiperdocumento é um documento (ou aplicação) manipulado por um sistema de hipertexto e possui três elementos básicos:

- nós, que contém a informação pura;
- ligações ou "links", que conectam um nó ao outro, determinando a estrutura do hiperdocumento; e
- âncoras, que são a representação visual de uma ligação e determinam os pontos através dos quais o usuário irá interagir e recuperar as informações armazenadas em outros nós.

Como já foi dito, uma das funções de um sistema de hipertextos é a autoria de hiperdocumentos, ou seja, a inserção de nós de informação na hiperbase e a definição da rede que eles formarão através do estabelecimento de ligações entre esses nós. A autoria de hiperdocumentos pode ser classificada em duas categorias [Gar93]: autoria em ponto pequeno, que trata do preenchimento dos nós com a informação desejada; e autoria em ponto grande, que trata da definição da estrutura da rede formada pelos nós e de como o usuário poderá navegar através dessa rede. Nesse sentido, este trabalho trata da solução de problemas relacionados com a autoria de hiperdocumentos em ponto grande.

A comunicação apoiada por computador também deu origem a outras formas de apresentação de informação além do hipertexto. Uma dessas formas é a integração de dados dinâmicos e estáticos apresentados através de mídias diferentes e percebidos por sentidos humanos também diferentes, explorando, principalmente, as relações temporais entre as porções de informação. Essa forma de expressão de informação é conhecida como multimídia. Com aplicações multimídia é possível apresentar dados de naturezas diferentes de forma concorrente e sincronizada, como áudio e vídeo, ou áudio e imagens. A exploração mais proveitosa dos sentidos humanos dá às aplicações multimídia um grande poder de expressão, o qual vem sendo vastamente utilizado nas aplicações atuais de sistemas de informação.

A limitação de somente conter informação textual foi logo modificada nos hipertextos e foram incorporados outros tipos de informação, apresentados através de várias mídias, como som, gráficos e imagens, o que iniciou uma aproximação entre as abordagens de hipertexto e multimídia. O enriquecimento das formas de comunicação obtido através dessas duas formas de expressão da informação é o ponto de partida para a concepção de um novo conceito: a hiperídia, a integração de hipertexto e multimídia em um só modelo de comunicação. Os hiperdocumentos, além de serem organizados de forma não linear e suportarem a navegação, passam a conter informações multimídia, constituindo uma forma ainda mais poderosa de expressão de informação. Os hiperdocumentos contidos na WWW são, atualmente o principal exemplo de aplicações hiperídia. Um detalhe interessante sobre a WWW é que, além da grande utilização de recursos multimídia, há uma distribuição da hiperbase pela Internet.

Os sistemas hiperídia e, conseqüentemente, os hiperdocumentos, são produtos de software e como tais, segundo os princípios da engenharia de software, demandam modelos que sistematizem e suportem a automatização de seu desenvolvimento. Com o aumento da dimensão e da demanda por aplicações hiperídia, nota-se a necessidade de modelos voltados à especificação dessas aplicações, favorecendo a melhoria, principalmente, da usabilidade e da manutenibilidade dos hiperdocumentos.

Nesse sentido, Garzotto et al. [Gar93] identificam quatro abordagens para a modelagem da estrutura de hipertextos: orientadas a aplicações, que modelam explicitamente a semântica do domínio de aplicação, que incluem o modelo de referência Dexter [Hal90] e o modelo baseado em teoria dos conjuntos de Garg [Gar88], tentando identificar as abstrações relevantes encontradas em vários sistemas existentes; os modelos comportamentais — preocupados com a modelagem do comportamento dinâmico associado com redes de hipertexto e com a semântica de navegação através da rede —, como o modelo Trellis [Sto89], baseado em Redes de Petri, o modelo HMBS [Oli95], baseado em statecharts, e o modelo de Tompa [Tom89], baseado em hipergrafos, e os modelos implementacionais, que enfatizam estruturas topológicas como blocos de construção para criar redes de hipertextos, tais como o Hypercard [App94], Guide [Bro87] e KMS [Aks88].

Os objetivos deste trabalho, descritos na próxima subseção, envolvem a extensão do modelo HMBS, classificado como um modelo comportamental de hiperdocumentos.

1.2 Objetivos

O HMBS (“Hypertext Model Based on Statecharts”) [Oli95] é um modelo de especificação de hiperdocumentos que, como pode ser visto nos próximos capítulos, não é

adequado para especificar requisitos temporais e de sincronização relativos à apresentação de informações multimídia. No entanto, a satisfação desses requisitos é essencial para qualquer modelo de especificação de aplicações hiperfídia, visto que o domínio dessas aplicações abrange apresentações de dados multimídia, além de dados puramente estáticos.

Identificado o potencial do modelo HMBS, devido ao uso de statecharts como técnica subjacente, o que garante ao modelo um grande poder de especificação de apresentações concorrentes de dados e constitui um passo na direção da especificação de multimídia, nota-se que o modelo dá margem a extensões para modelar aplicações hiperfídia. Além disso, como a definição de statecharts utilizada pelo HMBS é um subconjunto da definição original, em que se encontram algumas características que permitem, com alguma limitação, especificar requisitos de sincronização, nota-se também que extensões aos statecharts podem solucionar alguns dos problemas do modelo.

A literatura, por sua vez, conta com vários modelos de especificação de aplicações multimídia e hiperfídia, muitos deles baseados em redes de Petri, uma técnica com características similares às de statecharts, indicando um caminho para adaptações de soluções de uma técnica para outra, o que, em termos práticos, significa que alguns esforços podem ser poupados no trabalho de extensão do modelo HMBS.

Diante desses argumentos, este trabalho apresenta uma forma de especificar aplicações hiperfídia com requisitos temporais e de sincronização complexos através de um modelo baseado no HMBS e, portanto, em statecharts. O modelo apresentado permite a especificação de requisitos relacionados com a apresentação de informações multimídia em hiperdocumentos, além de outras características antes não suportadas.

1.3 Organização

Esta dissertação está organizada da seguinte forma: o presente capítulo introduz o problema e apresenta os objetivos do trabalho. O capítulo 2 apresenta uma classificação dos requisitos de aplicações hiperfídia. O capítulo 3 apresenta uma visão do estado da arte na área de especificação de aplicações de multimídia e hiperfídia. O capítulo 4 apresenta uma linguagem de alto nível, baseada em statecharts, para a especificação de hiperdocumentos: os *hypercharts*; o capítulo 5 apresenta o modelo XHMBS, juntamente com exemplos de notação e uso do novo modelo. Ao final desse capítulo, o modelo é comparado com os principais modelos apresentados no capítulo 3 e confrontado com os requisitos de aplicações hiperfídia apresentados no capítulo 2. Finalmente, as conclusões deste trabalho, assim como as perspectivas para a sua continuidade, são apresentadas no capítulo 6.

No apêndice A é apresentado um modelo de objetos que constitui um ponto de partida para a especificação de um sistema de apoio ao XHMBS. No apêndice B, o exemplo especificado no capítulo 5 é apresentado em uma versão alternativa, ilustrando uma das abordagens notacionais do modelo proposto.

Capítulo 2

Classificação dos requisitos de aplicações hipermídia

2.1 Considerações iniciais

Os requisitos de aplicações hipermídia são aqueles encontrados nos domínios das aplicações multimídia e dos hipertextos, uma vez que o conceito de hipermídia tem origem na união dos conceitos de multimídia e hipertexto. No contexto deste trabalho, como uma compilação do que se pode encontrar na literatura, esses requisitos são divididos em cinco grupos principais: requisitos relativos à obtenção e organização da informação; requisitos relativos à sincronização da apresentação; requisitos relativos à interação entre usuário e computador; requisitos relativos à apresentação da informação e requisitos relativos à navegação do hiperdocumento.

2.2 Requisitos relativos à obtenção e organização da informação

Os requisitos relativos à obtenção e organização da informação das aplicações hipermídia correspondem às necessidades do autor de gerar documentos e organizar a informação da melhor forma possível. Nesse sentido, um modelo de especificação de hiperdocumentos deveria permitir a identificação das relações intrínsecas do domínio da aplicação, possibilitando a definição de componentes do hiperdocumento de forma mais imediata. Além disso, espera-se que o modelo permita que o autor estruture a informação de forma conveniente e que seja possível abordar o problema de autoria em vários níveis de abstração e granularidade.

Os principais requisitos relativos à obtenção e organização da informação de aplicações hipermídia encontrados na literatura são descritos a seguir.

2.2.1 Modelagem do domínio da aplicação

Esse requisito está relacionado com a modelagem lógica ou semântica de elementos do mundo real, os quais podem dar origem a componentes ou classes de componentes da aplicação. Modelar o domínio de uma aplicação é o primeiro passo para se obter, de modo sistemático, uma aplicação automatizada consistente com a realidade. Normalmente, modelos com essa capacidade são extensões de outros modelos com origem na área de modelagem de dados, como o modelo Entidade-Relacionamento, por exemplo. HDM [Gar93] e RMM [Isa95] são exemplos de abordagens cujos focos estão voltados para a modelagem do domínio da aplicação.

2.2.2 Derivação de elementos da aplicação

Fortemente associado com a modelagem do domínio, esse requisito está relacionado com as facilidades que um modelo pode oferecer para se identificar classes de elementos de hipermídia que surgem em decorrência da criação de outros elementos ou devido a relações de domínio entre as classes de informação existentes. Possivelmente, a manifestação mais

importante desse requisito em sistemas de apoio à autoria de aplicações hipermídia seja a derivação automática de ligações entre nós de hiperdocumentos, visto que o estabelecimento de ligações é um problema complexo e que pode comprometer a navegação pela informação. A derivação de componentes também é um dos focos das abordagens HDM e RMM.

2.2.3 Estruturação da informação

Em aplicações hipermídia a informação pode ser organizada de três formas básicas: hierárquica; linear; ou em rede [Gin95]. Essas formas de organização podem ser exploradas pela semântica de navegação do modelo, que provê a maneira pela qual a informação será apresentada. Espera-se que um modelo de especificação permita a representação dessas três formas de estruturação de forma conveniente.

2.2.4 Separação entre estrutura e conteúdo da informação

Quando se cria um modelo de uma aplicação hipermídia, seja ela instanciável com informações de uma hiperbase abrangente ou um hiperdocumento com informações previamente conhecidas, duas das principais tarefas a serem conduzidas são a estruturação da informação a ser utilizada e a distribuição do conteúdo dessa informação pela estrutura definida. É importante que a estrutura da informação, que, na prática, define a estrutura da própria aplicação, esteja separada do seu conteúdo. Assim, uma modificação não-semântica no conteúdo da informação não deverá demandar alterações sobre a estrutura definida anteriormente. Um mecanismo de modelagem frequentemente utilizado pelos modelos de especificação para a satisfação desse requisito é o mapeamento entre unidades estruturais e unidades de informação. Essa abordagem é adotada por modelos como o Trellis [Sto89] e o HMBS [Oli95], entre outros. Outras propostas, como a do modelo Dexter [Hal90], que é um modelo de referência, sugerem uma divisão da especificação em níveis, com o conteúdo da informação alocado em um nível abaixo daquele em que se encontra a especificação da estrutura.

A idéia de separar a estrutura e o conteúdo da informação não corresponde, absolutamente, à anulação das relações estruturais identificadas na informação durante uma eventual fase de modelagem. Assim, modificações semânticas no conteúdo da informação poderão demandar alterações estruturais, as quais deveriam se refletir nos modelos da aplicação, então modificados em nível de análise. De qualquer forma, é desejável que os modelos de especificação possuam mecanismos que permitam a satisfação desse requisito.

2.2.5 Várias visões da mesma informação

Uma aplicação pode necessitar de que certas porções de informação sejam vistas de maneiras diferentes. Por exemplo: uma legenda de um vídeo pode ser apresentada em língua inglesa ou portuguesa, dependendo do usuário que está assistindo à apresentação. Um modelo de especificação pode oferecer recursos que permitam a satisfação desse requisito com o menor esforço possível. Separar a estrutura do conteúdo da informação na especificação das aplicações pode ser um passo importante nesse sentido.

2.2.6 Versões “sob medida” do mesmo hiperdocumento

Uma vez criado ou instanciado, um hiperdocumento poderá ser lido (ou navegado) por diversos usuários, muitos deles com necessidades diferentes, de forma que pode ser interessante que o mesmo hiperdocumento seja personalizado para certos usuários, com certas informações adicionais sendo apresentadas, ou com a omissão de outras porções de informação. Uma utilidade da personalização de hiperdocumentos é o controle de acesso a determinadas porções de informação, bastante demandado em sistemas de informação, principalmente os baseados no recente conceito de “intra-nets”.

2.2.7 Especificação em vários níveis de abstração

Um modelo de especificação de aplicações hipermédia, como outros modelos voltados para outros tipos de software, deve permitir que a tarefa de especificação seja abordada em níveis diferentes de abstração, o que facilita a tarefa de decomposição do problema.

2.2.8 Especificação em vários níveis de granularidade

Além de permitir que o problema seja abordado em níveis de abstração diferentes, no caso das aplicações hipermédia, a especificação de certas características, como controle de sincronização, exige do modelo a capacidade de especificar os componentes em níveis de granularidade também variáveis. Isto é, pode-se especificar, por exemplo, a sincronização entre dois objetos, como elementos atômicos ou pode-se abordar o problema considerando cada objeto como uma composição de várias unidades, o que permite que o controle de sincronização seja aplicado sobre essas porções menores. Assim, o nível de granularidade da especificação pode ser mais fino ou mais grosso, conforme a natureza dos elementos envolvidos e a necessidade de representação do desenvolvedor.

Em aplicações hipermédia, o nível de granularidade é determinante quando se aborda o problema da aplicação de mecanismos de sincronização nas porções multimídia dos hiperdocumentos. Pode-se dizer que esse requisito está bastante relacionado com requisito relativo ao nível de abstração da especificação, visto que, para se especificar o controle de sincronização em níveis mais finos de granularidade, é necessário que o nível de abstração seja menor. No entanto, deve-se notar que é possível que, mesmo com um nível baixo de abstração, o controle de sincronização seja especificado com granularidade grossa.

2.3 Requisitos relativos à sincronização da apresentação

Os requisitos relativos à sincronização da apresentação dizem respeito aos relacionamentos temporais entre as porções de informação que compõem uma aplicação. Além disso, esses requisitos envolvem componentes do sistema que, de alguma forma, possam causar alterações nos relacionamentos temporais especificados pelo autor da aplicação. Os requisitos de sincronização de uma aplicação hipermédia estão fortemente relacionados com o seu domínio multimídia. Assim, quanto mais recursos multimídia forem exigidos de um hiperdocumento, mais requisitos relativos à sincronização deverão ser satisfeitos.

Quando se pretende especificar os relacionamentos de sincronização em aplicações hipermédia, um elemento importante deve ser definido: a cadeia de dados. Uma cadeia de dados é uma coleção de dados de uma única mídia, utilizada em uma porção contínua da

apresentação, com origem em uma única fonte. As cadeias de dados são chamadas por alguns autores de objetos de mídia [Little90, Woo94].

Uma aplicação hipermídia pode ser distribuída ou não-distribuída. Basicamente, diz-se que, quando a aplicação é distribuída, os locais de origem e de destino das cadeias de dados não coincidem, ou seja, as cadeias de dados devem ser supridas através de uma rede de comunicação. Por outro lado, quando os locais de origem e de destino das cadeias de dados coincidem, diz-se que a aplicação é não-distribuída ou local. O fato de uma aplicação ser distribuída faz com que existam mais requisitos relativos à sincronização do que quando a aplicação é local, uma vez que a existência de um sistema de comunicação subjacente adiciona elementos que podem influenciar na sincronização da apresentação da informação. Para este trabalho, interessam todos os requisitos, de forma que, genericamente, todos os requisitos descritos são relativos a aplicações distribuídas. Muitos dos requisitos apresentados nesta seção foram adaptados da classificação apresentada em [Coo95] para aplicações multimídia.

Dessa forma, os requisitos relativos à sincronização podem ser divididos em dois subgrupos: os relativos à cadeia de dados e os relativos à comunicação. Há, ainda, requisitos relativos à obtenção e organização da informação e à interação entre usuário e computador que podem influenciar as relações de sincronização de uma aplicação.

2.3.1 Requisitos de sincronização relativos à cadeia de dados

Uma vez definida a noção de cadeia de dados e estabelecido o propósito de se descrever os requisitos relativos às aplicações distribuídas, pode-se descrever aqueles requisitos que surgem da necessidade de se gerar, transportar, obter e apresentar uma cadeia de dados.

2.3.1.1 Local de origem

Um local de origem é uma porção do sistema em que as cadeias de dados são providas. Um local de origem não é necessariamente equivalente a um local físico, ou seja, pode ser necessário que vários locais físicos sejam utilizados para gerar uma única cadeia de dados. No entanto, a sincronização das cadeias de dados só é realizada nos locais de destino. Isso quer dizer que a adição de locais de origem à aplicação não corresponde ao aumento da complexidade da especificação. Uma especificação deve conter as informações necessárias sobre os lugares de origem das cadeias de dados, incluindo suas propriedades de distribuição, armazenamento e geração.

2.3.1.2 Local de destino

O local de destino de uma cadeia de dados é o único local físico capaz de apresentá-la. Um local de destino deve possuir todas as características técnicas necessárias à apresentação da cadeia de dados com a qualidade desejada pelo autor. Para o caso das aplicações distribuídas, a existência de mais de um local de destino para uma cadeia de dados exige que todas as cadeias obedeçam à mesma relação temporal através de toda a rede. Dessa forma, diz-se que a adição de locais de destino à aplicação aumenta a complexidade da especificação. Assim como deve ocorrer com a especificação de locais de origem, um modelo deve permitir a especificação dos locais de destino das cadeias de dados.

2.3.1.3 Intervalos de apresentação

As cadeias de dados, ou porções dessas cadeias, devem ser sincronizadas segundo intervalos de apresentação, isto é, uma cadeia de dados, segundo esse requisito, pode ser tratada como a sua duração, ou seja, um intervalo de tempo. A literatura concorda com a notação de intervalos temporais [All83], que classifica os treze possíveis relacionamentos temporais entre quaisquer dois intervalos. A figura 2.1 mostra os sete relacionamentos temporais básicos entre dois intervalos. A partir desses sete relacionamentos é possível obter, por inversão, os outros seis. O relacionamento de igualdade não possui inverso, ou então, quando invertido, resulta nele próprio.

Quando as aplicações envolvem sincronização entre diferentes cadeias de dados, é necessário que o modelo de especificação seja capaz de representar essas relações temporais, se possível formalmente, para que se garanta a precisão da sincronização.

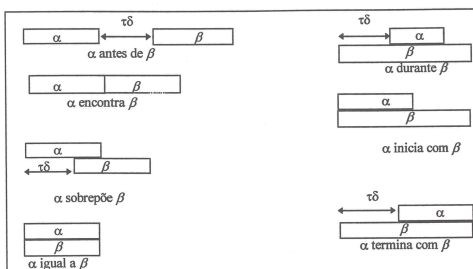


Figura 2.1 - Relacionamentos básicos entre intervalos temporais

2.3.1.4 Tipos de fonte

O local de origem de uma cadeia de dados pode fornecer os dados de duas formas: pré-orquestrada ou em tempo real. São exemplos de fontes pré-orquestradas os CD-ROMs e os discos rígidos das máquinas de origem. Uma fonte de tempo real pode ser uma câmera de vídeo que fornece imagens para a aplicação em tempo de execução. No caso das fontes de tempo real, o autor não tem como exercer controle de sincronização sobre a cadeia de dados, ficando limitado somente a iniciar e interromper a apresentação dos dados. Juntamente com a especificação do local de origem das cadeias de dados, um modelo deve permitir a especificação do tipo de fonte que esse local de origem constitui.

2.3.1.5 Dependência temporal

Uma cadeia de dados pode ser independente de tempo (discreta), ou dependente de tempo (contínua). Uma cadeia de dados discreta é aquela cuja apresentação é considerada completa tão logo ela tenha sido exibida, como imagens estáticas e textos, por exemplo. Já as cadeias de dados contínuas são aquelas que necessitam de uma apresentação contínua dos

dados e são consideradas completas somente quando o último dado é apresentado. Trechos de vídeo e áudio são os exemplos mais conhecidos de cadeias de dados contínuas. A especificação da apresentação e do controle de sincronização de cadeias discretas e contínuas pode ser bastante diferente. A especificação desse requisito pode ser omitida, uma vez que a dependência ou não do tempo irá se refletir nos outros requisitos da apresentação.

2.3.2 Requisitos de sincronização relativos à comunicação

Os requisitos relativos à comunicação estão fortemente relacionados com a descrição dos parâmetros de qualidade de serviço exigidos do sistema de comunicação subjacente. Esses requisitos determinam as taxas de funcionamento da rede de comunicação de forma que os atrasos toleráveis sejam respeitados e que as relações de sincronização determinadas em tempo de autoria sejam suportadas durante a apresentação da informação no local de destino.

2.3.2.1 Atraso

O requisito mais conhecido entre os de comunicação é o atraso ("delay"). Para as aplicações hipermédia, o atraso indica quanto tempo o sistema deve esperar pela entrega de uma cadeia de dados, ou parte dela, no local de destino. Adicionalmente, é necessário especificar, normalmente através de re-sincronização, o que deve ser feito caso o atraso tolerável não seja respeitado. Entre os atrasos que devem ser considerados, estão aqueles gerados pelo processamento das cadeias de dados, como compressão e criptografiação.

2.3.2.2 "Jitter"

Uma vez estabelecido o intervalo de tempo entre as entregas das cadeias de dados, ou o atraso entre elas, é possível, em um nível de granularidade mais fino, especificar a variação tolerável desse intervalo, isto é, quão antes ou quão depois do momento ideal o sistema ainda poderá apresentar os dados no local de destino sem causar um erro na aplicação. Essa variação nos atrasos é conhecida como "jitter" e o seu efeito é a inserção ou remoção de espaços na apresentação de uma cadeia de dados.

2.3.2.3 Taxas de "throughput"

Para apresentar as cadeias de dados de forma correta, é necessário que o sistema trabalhe com taxas de desempenho conhecidas. As taxas de "throughput" determinam a habilidade do sistema em processar e apresentar os dados. Esse requisito pode ser derivado das especificações do atraso e do "jitter", sendo especificado indiretamente.

2.3.2.4 Sensibilidade a erros

Finalmente, derivado dos outros requisitos de comunicação e da dependência temporal das cadeias de dados, o último requisito de comunicação a ser considerado é a sensibilidade a erros de transmissão. Sabe-se que mídias contínuas exigem um tráfego mais intenso da rede e que são menos sensíveis aos erros de transmissão, dado que a necessidade é de uma apresentação contínua e não de precisão absoluta. Por outro lado, as mídias discretas são mais sensíveis a erros de transmissão e não exigem muito controle de tráfego da rede. Essas características devem ser especificadas para que a apresentação corresponda à qualidade

esperada pelo autor. Também é necessário especificar que tipo de medidas deverão ser tomadas se ocorrerem erros considerados inaceitáveis pela aplicação.

2.4 Requisitos relativos à interação entre usuário e computador

As aplicações hipermédia caracterizam-se por serem interativas. Dessa forma, os requisitos relativos à interação usuário-computador devem ser suportados pelos modelos de especificação.

Basicamente, o que se espera dos modelos de especificação é que seja possível, além de simplesmente explicitar quais serão as possíveis interações em um determinado ponto da apresentação, também especificar como será a re-sincronização da apresentação e se o usuário ficará desorientado quando interagir com o hiperdocumento. Dessa forma, em algumas classificações, os requisitos relativos à interação entre usuário e computador podem ser considerados um subconjunto dos requisitos relativos à sincronização.

As interações a serem especificadas em uma aplicação hipermédia podem ser classificadas como interações de multimídia e interações de navegação. Os requisitos relativos às interações de navegação são os que correspondem à semântica de navegação proposta pelo modelo de especificação, enquanto que os relativos às interações de multimídia são aqueles relacionados com os controles de um painel de um "CD-Player" ou videocassete, como "fast-forward/rewind", "congelar", "descongelar", "alterar a velocidade", entre outros. Um modelo de especificação, normalmente, oferece um conjunto de interações possíveis para serem utilizadas. É possível, no entanto, que um modelo dê liberdade ao autor para que ele escolha quais interações serão possíveis em uma apresentação e até mesmo que novas interações sejam criadas em tempo de autoria.

2.5 Requisitos relativos à apresentação da informação

Os requisitos relativos à apresentação da informação dizem respeito à forma com que a informação será apresentada e à qualidade dessa apresentação.

2.5.1 Coordenação espacial

O primeiro requisito desse grupo é a coordenação espacial da informação. É necessário que o modelo de especificação seja capaz de especificar claramente a disposição espacial da informação a ser apresentada. Nas apresentações multimídia, várias cadeias são apresentadas ao mesmo tempo e podem alternar suas posições, como em uma animação, por exemplo. Os problemas a serem resolvidos variam desde a simples alocação e liberação de uma porção do dispositivo de visualização até o controle de choques entre alocações de recursos concorrentes ou sucessivas.

2.5.2 Controle de atributos globais da apresentação

O controle dos atributos globais da apresentação também constitui um requisito importante. Há alguns atributos que dizem mais respeito à apresentação como um todo do que à cadeia de dados, como o volume do áudio, a fonte dos textos e o padrão de cores das imagens, entre outros [Har94]. Esse requisito está relacionado com interações que o usuário

pode realizar para atualizar o valor desses atributos. Um modelo de especificação deve oferecer formas convenientes para especificar requisitos desse tipo.

2.5.3 Tipo de mídia da cadeia de dados

O tipo de mídia das cadeias de dados é o requisito que determina o tipo de dados que será apresentado. Esse requisito é de vital importância para a apresentação da informação, visto que, sem essa especificação, não é possível alocar os recursos corretos do sistema de apresentação.

2.6 Requisitos relativos à navegação do hiperdocumento

Uma vez gerado, seja através da instanciação de um esquema de aplicação, ou diretamente, com dados já conhecidos, o hiperdocumento pode ser utilizado. A especificação da navegação do hiperdocumento envolve a determinação do comportamento dinâmico da apresentação quando as ligações entre componentes são seguidas.

2.6.1 Noção de visita a um nó

A noção mais importante a ser suportada por um hiperdocumento é a de visita a um nó. Por visita entende-se o conjunto de efeitos áudio-visuais a que o usuário é submetido [Har94]. Diante de um novo conceito em hipermídia, o de hipermídia ativa [Sto90], diz-se que o tempo de uma visita pode ser definido pelo usuário em tempo de navegação ou especificado pelo autor através do modelo de especificação, ou seja, o usuário poderá seguir uma ligação a partir de um certo nó do hiperdocumento, determinando o final daquela visita, ou a própria aplicação poderá fazê-lo, devido à passagem de uma quantidade de tempo pré-definida.

2.6.2 Estruturas alternativas de acesso à informação

É possível que haja diferentes formas de acessar as informações além de seguir as ligações disponíveis em um hiperdocumento. O início da navegação pode se dar através de índices fornecidos pelo sistema de apresentação, ou até mesmo através de mecanismos de pesquisa por assunto ou palavras-chave. Normalmente, essa característica está mais associada aos sistemas do que às aplicações. No entanto, se o modelo de especificação suportar esses mecanismos de acesso, sua implementação em um sistema de apoio será facilitada.

2.6.3 Ligações temporais

Podem existir hiperdocumentos com ligações que são seguidas automaticamente, caso o usuário não escolha alguma específica dentro de um intervalo de tempo determinado. Tais ligações são denominadas ligações temporais e constituem um requisito importante para o que vem sendo chamado de hipermídia ativa. Dessa forma, um modelo atualizado poderia suportar esse requisito além das ligações convencionais.

2.6.4 Ligações dinâmicas

Quando está navegando por um hiperdocumento, o usuário pode utilizar mecanismos de pesquisa e o sistema de apresentação poderá gerar ligações não especificadas em tempo de

autoria. Essas ligações são conhecidas como dinâmicas e são características de aplicações que permitem formas alternativas de acesso à informação. Assim, os mecanismos de acesso providos pelo modelo podem colaborar para a satisfação desse requisito.

2.6.5 Ligações dentro de componentes

As ligações entre componentes são a forma mais comum de conexão entre porções de informação em aplicações hiperídia. No entanto, é possível que algumas aplicações necessitem de ligações dentro de componentes. Esse requisito pode ser excluído das aplicações de hipertexto, uma vez que qualquer texto pode ser dividido em dois componentes e uma ligação pode ser definida entre eles. Quando as ligações são necessárias dentro de cadeias de dados de outras mídias, como vídeo, por exemplo, a tarefa de divisão pode não ser trivial [Har94] e, nesse caso, é preciso que se defina uma ligação dentro do próprio componente. Um modelo de especificação deveria suportar esse requisito de forma simples e conveniente.

2.6.6 Contextos de informação

Quando uma ligação é seguida, é preciso que se especifique com toda clareza quais componentes deixarão de ser apresentados e quais passarão a estar visíveis. Esse requisito é chamado de contexto de informação ou contexto de ligação [Har94]. A satisfação desse requisito tem influência direta sobre a qualidade da navegação em termos de condução do usuário através das informações. Quanto melhor definidos os contextos de informação, menores serão as manifestações de fenômenos de desorientação por parte do usuário. Assim, é imprescindível que um bom modelo suporte esse requisito.

2.6.7 Visão da hierarquia do hiperdocumento

A desorientação do usuário pode ser reduzida através da satisfação de outro requisito importante para aplicações hiperídia: a possibilidade de que o usuário visualize a organização hierárquica da informação. Com isso, o usuário pode ter uma visão mais global da navegação e se orientar melhor para seguir as ligações. Os modelos de especificação conhecidos atualmente permitem, de alguma forma, que o autor inclua informações a respeito da hierarquia em um hiperdocumento, visto que essas informações podem ser tratadas como componentes comuns. A questão é que, dessa forma, essa é uma tarefa alocada exclusivamente para o autor, sem que o modelo possua qualquer mecanismo dedicado à especificação desse requisito. O ideal é que os modelos tratem esse requisito de forma específica e, assim, apoiem de forma mais conveniente o processo de autoria.

2.7 Considerações finais

Com a classificação de requisitos de aplicações hiperídia apresentada neste capítulo, torna-se possível o estabelecimento de um conjunto de parâmetros que pode ser levado em consideração quando da análise de modelos e ferramentas voltados para a especificação e implementação desse tipo de aplicação. Dessa forma, no contexto deste trabalho, as avaliações de propostas da literatura, assim como a condução das extensões realizadas sobre o HMBS até se chegar ao XHMBS, têm por base essa classificação.

Outras classificações de requisitos são consideradas na literatura, as quais levam em conta alguns aspectos mais específicos de determinados domínios de aplicação. A classificação apresentada neste trabalho tem o objetivo de abranger os conceitos apresentados em outras classificações de requisitos, constituindo uma referência genérica para o assunto no contexto deste trabalho. É certo que para discussões mais detalhadas acerca de determinados grupos de requisitos pode-se utilizar outras classificações, as quais, espera-se, poderão ser interpretadas segundo os termos e conceitos apresentados neste capítulo.

Capítulo 3

Modelos para especificação de aplicações de multimídia, hipertexto e hiperídia

3.1 Considerações iniciais

Para um melhor entendimento da problemática que existe em torno da área de hiperídia, é interessante definir os principais termos envolvidos. Em [Har94] é apresentada uma revisão em alto nível dos conceitos de hipertexto, multimídia e hiperídia, além de algumas considerações a respeito da integração de hipertexto e multimídia em aplicações de hiperídia.

Um hipertexto é modelado como uma rede de componentes relacionados através de ligações, ou "links". Os componentes interligados em um hipertexto são chamados de nós. As informações em cada nó podem ser apresentadas através de mídias diferentes, ou seja, é permitido que um nó contenha texto e que outro contenha áudio, por exemplo. Independentemente de todas as possíveis diferenças de implementação, qualquer hipertexto deve suportar a noção de visita a um nó por um tempo determinado pelo usuário, ou pelo autor da aplicação, em casos mais sofisticados. Em hipertexto, os efeitos audio-visuais de uma visita são considerados uma propriedade interna dos dados e não podem, portanto ser manipulados pelo autor da aplicação. A figura 3.1 ilustra, conceitualmente, um hipertexto.

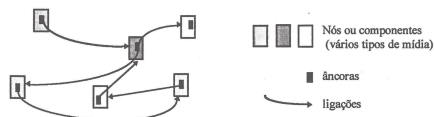


Figura 3.1 - Hipertexto (adaptado de [Har94])

Multimídia refere-se à integração de componentes de várias mídias diferentes, característica também encontrada em outros tipos de aplicações, como as de hipertexto. O fator diferencial em multimídia é a apresentação dos componentes em alguma ordem relativa, estabelecida pelo autor. Essa relação de ordenação depende de uma noção explícita de tempo, o que constitui um dos principais requisitos desse tipo de aplicação. Enquanto o usuário continua podendo ter controle sobre a seleção de quais componentes serão visitados, os componentes selecionados e apresentados podem mudar, isto é, sua apresentação pode progredir sem qualquer intervenção direta do usuário devido a essa noção de tempo.

Em termos de modelagem de aplicações, o que ocorre com apresentações multimídia é que há necessidade de que o modelo de especificação defina não somente os componentes da apresentação, mas também permita, entre outras coisas, a definição de relacionamentos de

ordenação temporal que especifiquem quando os componentes serão apresentados, com relação uns aos outros. A figura 3.2 ilustra uma aplicação multimídia conceitual.

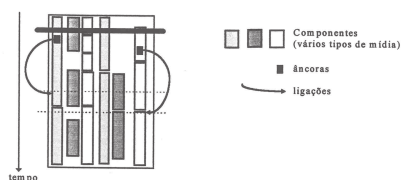


Figura 3.2 - Multimídia (adaptado de [Har94])

Os sistemas de apoio a aplicações multimídia suportam, normalmente, dois tipos de facilidades de navegação para que o usuário possa controlar uma apresentação: o ajuste da referência de tempo (indicada na figura pela linha horizontal mais espessa) e um tipo de ligação como as de hipertexto, indicada na figura pela âncora e pela seta de ligação. No primeiro caso, o usuário pode interagir com a apresentação através de interfaces de controle similares àquelas encontradas em videocassetes ou "CD-players". O usuário pode iniciar, parar, retroceder e avançar rapidamente na apresentação. No segundo caso, seguir uma ligação levaria o usuário a um ponto do tempo específico, determinado em tempo de autoria. Isso é similar a uma interação do tipo avanço (ou retrocesso) rápido ("fast forward / rewind") com a diferença de que, nesse caso, o ponto de parada é pré-determinado.

Hipermultimídia, nesse contexto, é a integração das características de hipertexto e multimídia em um só modelo. A figura 3.3 ilustra uma das formas de se combinar hipertexto e multimídia: fazendo com que os componentes do hipertexto contenham apresentações multimídia. Esse modelo considera duas características de hipermultimídia: a navegação hiperestruturada através do documento e a apresentação multimídia da informação. Essa abordagem, bastante simples, pode ser conveniente para vários tipos de aplicações hipermultimídia, mas pode se mostrar limitada em termos de flexibilização do processo de autoria e até mesmo de apresentação ao usuário.

O restante deste capítulo é dedicado a uma visão do estado da arte no que se refere à especificação de aplicações hipermultimídia. Devido ao fato de o HMBS [Oli95] ser um modelo adequado às características de hipertexto, mas limitado quanto às de multimídia, este trabalho procura revisar com maior ênfase os modelos voltados à especificação de aplicações multimídia, nas quais encontram-se os principais requisitos relativos à sincronização da apresentação da informação. Muitos dos modelos apresentados são dedicados somente a esse tipo de aplicação. Outros modelos são voltados somente a aplicações de hipertextos e outros, ainda, à integração das características de hipertexto e de multimídia. Os modelos revisados nas próximas subseções são apresentados sob a luz da classificação de requisitos de aplicações hipermultimídia apresentada no capítulo 2. Há de se notar a ênfase dada aos modelos formais de especificação, como os baseados em redes de Petri. Isso se deve ao fato de que os requisitos

de sincronização podem ser melhor especificados sob uma abordagem formal, dada a sua complexidade. Além disso, a abordagem do HMBS, modelo cuja extensão é apresentada neste trabalho, é formal e, assim, o foco de grande parte desta revisão é voltado para soluções baseadas nessa mesma abordagem.

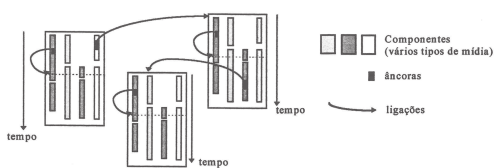


Figura 3.3 - Hipermídia (adaptado de [Har94])

3.2 Modelos baseados em Redes de Petri

As redes de Petri constituem uma técnica formal para a representação de sistemas caracterizados pela concorrência e sincronização de seu comportamento, como é o caso de sistemas reativos e de tempo real, entre os quais encontram-se as aplicações de multimídia e hipermídia.

Uma Rede de Petri é definida como uma tripla $PN = \langle T, P, N \rangle$ onde

$T = \{t_1, t_2, \dots, t_n\}$ é um conjunto de transições, representadas por barras.

$P = \{p_1, p_2, \dots, p_m\}$ é um conjunto de lugares, representados por círculos.

$A = \{T \times P\} \cup \{P \times T\} \rightarrow N$, é o mapeamento que representa os arcos que vão das transições para os lugares e dos lugares para as transições. Os arcos são representados por setas direcionadas.

Uma rede de Petri marcada é uma quádrupla $M_{PN} = \langle T, P, N, M \rangle$ que acrescenta uma marcação M que associa "tokens", ou fichas, a cada lugar da rede. As fichas são representadas por círculos cheios colocados dentro dos lugares. A marcação M pode ser definida como $M: P \rightarrow N$.

Diz-se que uma marcação da rede representa um dos seus estados específicos, ou seja, quais lugares estão ativos em um determinado momento. Através das marcações e da definição de uma semântica é possível simular (ou executar) uma rede de Petri. Graficamente, uma rede de Petri marcada possui a aparência e os elementos apresentados na figura 3.4.

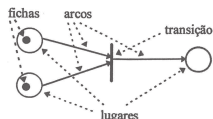


Figura 3. 4 - Rede de Petri marcada

Basicamente, a regra de disparo de uma transição em uma rede de Petri marcada diz que uma transição t está habilitada quando todos os seus lugares de entrada ($p \in P \mid \exists \langle p, t \rangle \in A \wedge t \in T$) estiverem ocupados por fichas. Nessas condições a transição t dispara instantaneamente, ocupando os lugares de saída ($p \in P \mid \exists \langle t, p \rangle \in A \wedge t \in T$) com fichas. Essa característica dificulta a especificação de comportamentos que não são instantâneos, como a exibição de um vídeo ou a apresentação de um trecho de áudio. Diante dessas dificuldades, alguns trabalhos têm sido desenvolvidos para estender as redes de Petri para que seja possível representar comportamentos temporais, principalmente em aplicações de multimídia.

As primeiras evoluções de redes de Petri foram tentativas de extensão dessa técnica para que ela pudesse ser utilizada para especificar aplicações de tempo real. A seguir são apresentadas brevemente as primeiras extensões temporais de redes de Petri.

3.2.1 "Timed Petri Nets"

As "Timed Petri Nets" [Ram74] foram a primeira tentativa de introduzir tempo em redes de Petri. Nessas redes, o disparo de uma transição consome um determinado tempo, de forma que a duração é associada às transições. Essa abordagem é bastante limitada e apresenta um problema conceitual: as transições não disparam instantaneamente. Dessa forma, durante o tempo em que a transição está disparando, o estado do sistema é indefinido. A figura 3.5-a apresenta um esquema de "Timed Petri Net".

3.2.2 "Time Petri Nets"

As "Time Petri Nets" [Mer74] associam, ao invés de uma duração, um intervalo de tempo às transições. Assim, o modelo permite a especificação de não-determinismo temporal e torna-se mais flexível do que as "Timed Petri Nets". O disparo das transições nessa variação da técnica é definido de forma mais consistente, uma vez que o estado do sistema é sempre conhecido. Isso se deve ao fato de que, tendo sido associado um intervalo $[x, y]$ a uma transição t , tem-se que t irá disparar em um momento entre x e y , de forma que, se o momento absoluto em que a transição tornou-se habilitada é dado por τ , o disparo de t não deverá ocorrer antes de $\tau + x$ e deverá ocorrer, no máximo, até $\tau + y$. Dessa forma, o disparo da transição é instantâneo. Um esquema de "Time Petri Nets" é apresentado na figura 3.5-b.

3.2.3 "Arc Time Petri Nets"

Uma variação importante das redes de Petri com notação temporal é o modelo "Arc Time Petri Net" [Wa83]. Nessa variação, os intervalos de tempo são associados aos arcos que entram nas transições. Com isso, a regra de disparo é alterada e exige que, além de as fichas estarem presentes em todos os lugares de entrada, o disparo ocorra no intervalo de tempo determinado pela intersecção dos intervalos contidos nos arcos de entrada. A figura 3.5-c apresenta um esquema de "Arc Time Petri Net".

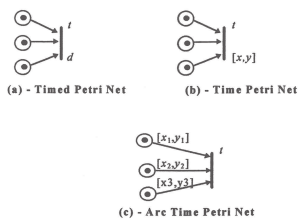


Figura 3.5 - Redes de Petri com notação temporal

Para essas três variações de redes de Petri são utilizadas definições que levam em consideração a utilização de redes de Petri ponderadas, ou seja, aquelas redes cujos arcos possuem um peso associado. Esse peso equivale ao número de fichas que um arco retira ou deposita em um lugar, respectivamente, de entrada ou de saída, com relação à transição. Nessas redes, a regra de disparo é alterada, exigindo-se que o número de fichas de cada lugar de entrada de uma transição seja maior ou igual ao peso do respectivo arco de entrada. Neste trabalho, serão consideradas, na maior parte dos casos, redes de Petri com peso igual a um, ou seja, cada arco de entrada retira somente uma ficha do lugar de entrada e cada arco de saída deposita somente uma ficha no lugar de saída, semântica que se aproxima melhor da de statecharts.

As três variações de redes de Petri apresentadas até este ponto são úteis, com algumas limitações, para especificar requisitos relativos à estruturação da informação, intervalos de apresentação e sincronização entre cadeias, além de poderem ser usadas para especificar a noção de visita a um nó e certos tipos de ligações temporais, disponibilizando uma semântica de navegação para hiperdocumentos. No entanto, a transformação dessas variações em modelos para especificação de aplicações de multimídia, hipertexto ou hiperfídia depende de outras definições para que os demais requisitos relativos a essas aplicações sejam satisfeitos. Nas subseções seguintes, são apresentados modelos que utilizam essas variações de redes de Petri como ponto de partida para a especificação de apresentações multimídia e hiperdocumentos.

3.2.4 O modelo OCPN

Little, em [Lit90], apresenta uma abordagem baseada em redes de Petri para a especificação de requisitos de sincronização de documentos multimídia sem maior preocupação com aspectos de interação com o usuário. O modelo proposto, chamado OCPN ("Object Composition Petri Net") foi um importante ponto de partida para a especificação formal de aplicações multimídia. O modelo aumenta as Redes de Petri através da associação de durações e recursos aos lugares. Dessa forma, uma OCPN é definida como segue:

$C_{OCPN} = \langle T, P, A, D, R, M \rangle$ onde:

$\langle T, P, A, M \rangle$ é uma rede de Petri marcada;

$D: P \rightarrow \mathbb{R}$ é um mapeamento do conjunto de lugares para números reais (durações).

$R: P \rightarrow \{r_1, r_2, \dots, r_k\}$ é um mapeamento do conjunto de lugares para os recursos.

O modelo permite, através do uso das durações e transições com mais do que um arco de entrada, especificar os requisitos de sincronização entre as cadeias de dados, denominadas objetos de mídia pelo modelo. Esses recursos possibilitam a composição de uma grande quantidade de apresentações multimídia.

A especificação das relações temporais entre os recursos associados aos lugares é baseada em intervalos temporais [All83] e a idéia adotada por esse modelo é herdada por todos os outros, subsequentemente, é a de representar essas relações temporais com OCPNs de forma que, através da combinação das pequenas redes obtidas, seja possível representar todo um documento, satisfazendo aos seus requisitos relativos aos intervalos de apresentação. Para conseguir essa representação, considera-se que α e β são processos de durações τ_α e τ_β , respectivamente. O intervalo τ_δ é uma diferença de tempo entre dois processos quaisquer. A figura 3.6 mostra como ficam as representações das relações temporais em OCPN.

As representações das relações temporais em OCPN, apresentadas na figura 3.6, podem ser unificadas em um modelo parametrizado, como o que é apresentado na figura 3.7.

O modelo OCPN é acompanhado de um esquema hierárquico de base de dados e de um algoritmo de recuperação de dados, tratando, portanto, da autoria, armazenamento e apresentação de documentos multimídia. Dessa forma, pode-se dizer que o modelo propõe soluções que vão além da especificação da aplicação, tratando de sua implementação. O modelo, estritamente voltado para a especificação de documentos multimídia, não especifica todos os requisitos relativos à obtenção e organização da informação. Não há, como dito anteriormente, preocupação com os requisitos relativos à interação com o usuário, nem com a apresentação da informação, o que poderia ser solucionado com a associação de informações espaciais aos recursos mapeados para os lugares. O modelo também não permite especificar requisitos de distribuição de objetos multimídia sobre um sistema descentralizado, como atrasos introduzidos por sistemas de comunicação e de armazenamento. Além disso, por se tratar de um modelo de especificação de aplicações multimídia, os requisitos relativos à navegação de hiperdocumentos não podem ser especificados com OCPN.

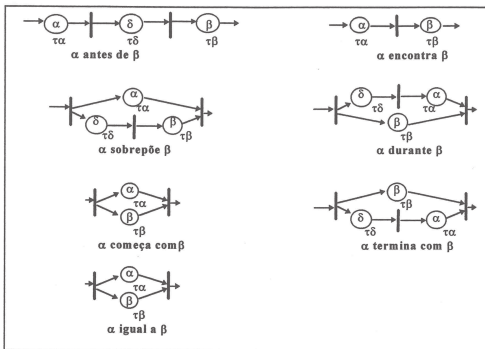
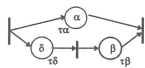


Figura 3.6 - Representação das relações temporais através de OCPN



Relação	Relacionamentos entre Parâmetros	Duração Global (τ_{TK})
α antes de β	$\tau\delta \neq 0$	$\tau_{TK} = \tau\alpha + \tau\beta + \tau\delta$
α encontra β	$\tau\delta = 0$	$\tau_{TK} \geq \tau\alpha + \tau\beta$
α sobrepõe β	$\tau\alpha < \tau\beta$ $\tau\delta \neq 0$	$\tau_{TK} = \tau\beta + \tau\delta$
α durante ⁻¹ β	$\tau\alpha > \tau\beta$ $\tau\delta \neq 0$	$\tau_{TK} = \tau\alpha$
α começa com β	$\tau\alpha < \tau\beta$ $\tau\delta = 0$	$\tau_{TK} = \tau\beta$
α termina ⁻¹ com β	$\tau\alpha = \tau\beta$ $\tau\delta \neq 0$	$\tau_{TK} = \tau\alpha$
α igual a β	$\tau\alpha = \tau\beta$ $\tau\delta = 0$	$\tau_{TK} = \tau\alpha$

Figura 3.7 - Relacionamentos temporais representados com OCPN

Esquemáticamente, o modelo OCPN estende a rede de Petri da forma apresentada na figura 3.8.

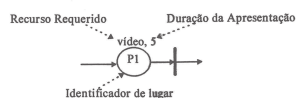


Figura 3.8 - Notação gráfica do modelo OCNP

3.2.5 O modelo Trellis

O modelo Trellis, em sua primeira versão [Sto89], utiliza as redes de Petri como uma extensão de grafos direcionados, bastante utilizados na especificação, principalmente da estrutura, de hiperdocumentos. A idéia do modelo é aproveitar a característica de grafo direcionado das redes de Petri e utilizá-las como mecanismo de especificação da estrutura ligada do hipertexto, satisfazendo os requisitos relativos à estruturação da informação. Além disso, o modelo utiliza uma outra característica das redes de Petri, a de autômato, para a especificação da semântica de navegação dos hiperdocumentos, satisfazendo, pelo menos, o requisito relativo à noção de visita a um nó.

O modelo Trellis não possui mecanismo para especificar os relacionamentos temporais que podem existir entre componentes dos hiperdocumentos, o que limita a especificação apenas a informações estáticas. Dessa forma, os requisitos relativos à sincronização da apresentação não podem ser completamente especificados.

Um hipertexto H , segundo esse modelo, é definido formalmente como segue:

$H = \langle N, C, W, B, P_1, P_d \rangle$, em que

$N = \langle T, P, A \rangle$ é uma estrutura de rede de Petri;

C é um conjunto de conteúdos do documento;

W é um conjunto de janelas;

B é um conjunto de botões;

P_1 é uma projeção lógica para o documento, responsável pelos mapeamentos que associam lugares a conteúdos e a janelas, bem como transições a botões.

P_d é uma projeção de display (física) para o documento que simplesmente associa as janelas e botões lógicos aos dispositivos físicos de apresentação.

Aumentando a rede de Petri subjacente com uma marcação (fazendo $N = \langle T, P, A, M \rangle$), tem-se que a semântica de execução da rede define a semântica de navegação do hiperdocumento. Assim, quando um lugar possuir fichas, o conteúdo associado a ele será colocado na janela lógica correspondente, a qual, através dos mapeamentos físicos definidos, será apresentada no dispositivo de apresentação. Devido às características de execução das redes de Petri, o modelo Trellis permite a especificação de apresentações e caminhos de

navegação concorrentes. A notação do modelo é fortemente baseada nos mapeamentos, de forma, que a rede de Petri subjacente não sofre alterações em sua aparência. Pode-se, no entanto, adotar a abordagem de nomear os lugares e as transições, respectivamente, com os nomes dos conteúdos e botões associados, o que facilita a leitura da especificação.

3.2.6 O modelo XOCPN

Seguindo a mesma linha de especificação temporal, outras abordagens de especificação baseadas em Redes de Petri procuram estender o modelo OCPN, cuja capacidade de representação é consideravelmente restrita. As novas propostas procuram cobrir deficiências como a impossibilidade de especificar interações com o usuário e o nível muito grosso de granularidade da especificação. Com isso, as extensões ao modelo OCPN possibilitam a especificação de aplicações interativas e de requisitos de sincronização para dados isócronos, isto é, dados contidos em cadeias que são apresentadas ao mesmo tempo.

O modelo XOCPN ("Extended Object Composition Petri Net") [Woo94] trata a deficiência relacionada com a granularidade encontrada no OCPN, permitindo que se especifique sincronização dentro das cadeias de dados e também entre cadeias cujas apresentações sejam isócronas. O modelo atinge esse objetivo através da divisão da informação em unidades menores denominadas SIUs ("Synchronization Interval Units"), formando uma cadeia no interior da qual são inseridos IPPs ("Interstream Pacing Points") em pontos pré-calculados. O modelo não deixa suficientemente claro como se dá o relacionamento entre IPPs de cadeias diferentes, mas é através desse mecanismo que se consegue especificar sincronização entre dados isócronos. Para a especificação de requisitos de sincronização entre cadeias de dados, ou objetos de mídia, o modelo lança mão de ISPs ("Inter-object Synchronization Points"), permitindo especificações em maior nível de granularidade. Todos esses pontos de sincronização são também representados por lugares da rede de Petri, dando origem a dois tipos de lugares: os de objetos e os de controle, o que faz aumentar o número de símbolos da especificação.

A sincronização entre cadeias é obtida através de políticas de bloqueamento e bloqueamento restrito. Com bloqueamento, a ação de apresentação pode ser suspensa até que a última SIU chegue ao local de destino. Com o bloqueamento restrito, essa ação fica bloqueada por um tempo específico, somente enquanto se espera que a última SIU chegue. Se ocorrer um "time-out", a última SIU armazenada é apresentada novamente, ignorando a SIU esperada. Essa política pode ser inapropriada para garantir a correção de alguns tipos de aplicação em que a continuidade da apresentação for desejada a qualquer custo.

O uso de lugares de controle permite que o desenvolvedor trabalhe com o conceito de canais virtuais – que representam os dispositivos de apresentação –, negociar parâmetros de qualidade de serviço com o sistema de comunicação subjacente, bem como tratar da alocação e liberação de recursos. Dessa forma, pode-se dizer que o modelo satisfaz aos requisitos de sincronização das aplicações hipermídia, mesmo sendo voltado somente para a especificação de documentos multimídia.

Outra característica importante no modelo XOCPN, e que faz com que as especificações geradas através dele sejam maiores, é a utilização de dois modelos, um referente ao transmissor e outro referente ao receptor. Esses modelos especificam, respectivamente, o

escalonamento de objetos multimídia e a sua apresentação com a manutenção dos requisitos de sincronização. A definição formal do modelo XOCPN é a seguinte:

$C_{XOCPN} = \langle T, P, A, D, Rs, M, Y, Z \rangle$ onde

$\langle T, P, A, M \rangle$ é uma rede de Petri Marcada,

$D: P \rightarrow (d_1, d_2)$ representa o mapeamento de lugares para o atraso antes de uma ação e a duração da ação, respectivamente.

$Rs: P \rightarrow (r_1, r_2, \dots, r_n)$ é o mapeamento de recursos de lugares para objetos.

$Y: P \rightarrow (Alocação_Recurso, Liberação_Recurso, Apresentação_SIU, Transmissão_SIU, Sincronização_entre_Cadeias)$ é um mapeamento de lugares para ações a serem realizadas durante a comunicação.

$Z: P \rightarrow (endereço_QoS, endereço_SIU, endereço_requis_sincronização)$ é um mapeamento de lugares para variáveis de implementação, as quais determinam os parâmetros de qualidade de serviço e os mecanismos de sincronização.

Graficamente, além do que se pode ver em uma especificação OCPN, a notação gráfica do modelo XOCPN possui a aparência da figura 3.9.

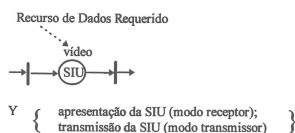


Figura 3.9 - Notação gráfica do modelo XOCPN

Apesar de todo o enriquecimento no que se refere às restrições de granularidade da especificação encontradas no OCPN, o modelo XOCPN ainda não permite a especificação de um dos requisitos relacionados com a comunicação, o "jitter", além de também não permitir a especificação de interações com o usuário.

3.2.7 O modelo DTPN

Os requisitos de interação com o usuário são parcialmente satisfeitos por uma outra extensão ao modelo OCPN, o DTPN ("Dynamic Timed Petri Nets") [Pra93], que define uma série de interações com o usuário comuns aos documentos multimídia, tais como "saltar", "apresentar em reverso", "congelar", "reiniciar" e "modificar a velocidade de apresentação". Por outro lado, o DTPN permanece com os problemas relativos à granularidade de especificação encontrados no OCPN e tratados no XOCPN.

A característica marcante do modelo DTPN é a existência de preempção e modificação das durações de apresentação das cadeias de dados, o que define o mecanismo de suporte às

interações com o usuário oferecidas. A duração da execução de um lugar da rede pode ser modificada temporariamente (pelo período em que aquele lugar estiver ativo), modificada permanentemente (passando a valer para todas as ativações daquele lugar), ou ainda terminada ou adiada. Com essas propriedades, a apresentação de um documento dá origem a uma classificação dinâmica das cadeias de dados, de acordo com qual cadeia recebe a interação do usuário. A cadeia que recebe a interação é chamada de cadeia primária, enquanto que as outras são chamadas de cadeias secundárias dependentes ou independentes, dependendo da relação de sincronização que tiverem com a cadeia primária.

Há um detalhe importante relacionado com as interações com usuários nesse modelo: a restrição imposta pela granularidade sobre a re-sincronização das cadeias de dados após uma interação. O ponto de início da apresentação do novo trecho, seja ele resultante de uma reversão da direção da apresentação ou de um salto, irá depender da granularidade da especificação, ou seja, a próxima unidade de apresentação a ser exibida será aquela que estiver associada ao próximo lugar da rede. Isso quer dizer que é impossível ao usuário saltar uma quantidade arbitrária de unidades de tempo na apresentação, ou mesmo fazer com que a apresentação do sentido inverso inicie exatamente no ponto onde ocorreu a interação. A base sobre a qual são modelados os mecanismos de interação com o usuário é uma decomposição hierárquica dos lugares da rede, de forma que um lugar é, na realidade, um bloco de construção definido por uma pequena OCPN, a qual possui sempre a mesma forma e tem as interações com o usuário associadas às suas transições. A definição formal do DTPN é a seguinte:

$$C_{DTPN} = \langle P, T, A, M, D, Rs, M, C, E \rangle \text{ onde}$$

$\langle P, T, A, M, D, Rs, M \rangle$ define uma OCPN

$C \subseteq P \times T$; $A \cap C = \emptyset$, onde C representa os arcos de escape.

$E: P \rightarrow \mathfrak{R}$ representa a duração restante como mapeamento do conjunto de lugares para o conjunto dos números reais. A relação E mantém a duração restante para a execução se o lugar for desativado e depois reativado. A relação D mantém a duração normal permitida. Inicialmente D e E têm o mesmo valor.

A figura 3.10 mostra um bloco de construção com as interações com usuário do modelo DTPN. A notação gráfica do modelo DTPN é a apresentada na figura 3.11.

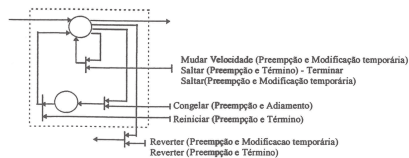


Figura 3.10 - Bloco de construção do modelo DTPN

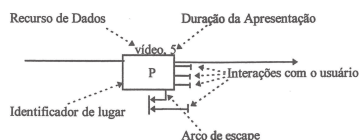


Figura 3.11 - Notação gráfica do modelo DTPN

3.2.8 O modelo TSPN

O modelo TSPN ("Time Stream Petri Net") [Dia93, Sen94] é uma outra extensão do OCPN em que os requisitos de interação com o usuário não podem ser especificados. As TSPN têm seu ponto forte na especificação de requisitos de sincronização com granularidade mais fina do que aquela proposta no modelo XOCPN. Na verdade, a granularidade da especificação em TSPN pode variar de acordo com a necessidade do especificador. Isso permite a especificação de uma quantidade maior de requisitos de sincronização, inclusive o "jitter".

O modelo representa requisitos de sincronização através da rotulação dos arcos que vão dos lugares para as transições. O rótulo é uma tupla $[\alpha, \eta, \beta]$ representando, respectivamente, o tempo mais adiantado de liberação da ficha para o disparo da transição, a duração nominal (ideal) da apresentação do recurso associado ao lugar, e o tempo mais atrasado de liberação da ficha. O uso dos parâmetros α e β permite a especificação do máximo "jitter" tolerável.

Nesse modelo os lugares representam somente pontos de processamento e as transições com mais de um lugar de entrada recebem um tipo e constituem o mecanismo de controle da estratégia de sincronização entre cadeias. A sincronização entre cadeias pode ser conseguida através de nove tipos de transição providos pelo modelo: "strong_or", "weak_and", "master", "or", "and", "strong-master", "weak-master", "or-master" e "and-master". Na estratégia "strong_or", a apresentação de uma das cadeias é suficiente para que a apresentação seja considerada completa, desde que as demais cadeias tenham sido apresentadas pelo tempo mínimo especificado nos respectivos arcos. Com a estratégia "weak_and" a apresentação só é considerada completa quando as apresentações de todas as cadeias forem completadas. Já na estratégia "master", exige-se apenas que uma cadeia, a "master", esteja completa para que a apresentação também seja considerada completa. O tipo "or" introduz a noção de indeterminismo de sincronização, fazendo com que a primeira cadeia a ser apresentada dispare a sincronização, independentemente das demais cadeias. O tipo "and" determina que todas as cadeias sejam apresentadas e que, caso isso não aconteça, a sincronização ocorra somente quando uma das cadeias tiver atingido seu tempo máximo especificado. Em tempo de apresentação são calculados intervalos dinâmicos de disparo para a transição tipada. As extremidades desses intervalos são consideradas de acordo com o tipo utilizado. No capítulo 4, onde este assunto é retomado, são apresentadas as formas gerais dos intervalos gerados por cada tipo de transição.

A definição formal do modelo TSPN é a que segue:

$C_{TSPN} = \langle P, T, B, F, M, IM, SYN \rangle$ onde

$\langle P, T, B, F, M \rangle$ define uma rede de Petri marcada ponderada, ou seja, uma rede de Petri onde o peso atribuído a um arco corresponde ao número de fichas que ele transporta.

se $A = \{a = \langle p, t \rangle \in P \times T \mid (B(\langle p, t \rangle) \neq 0)\}$ é o conjunto de arcos saindo dos lugares, então:

$IM: A \rightarrow Q^+ \times Q^+ \times (Q^+ \cup \infty)$ é o mapeamento de arcos que entram nas transições para números racionais positivos representando os valores da tupla $[\alpha, \eta, \beta]$. A duração representada por η nessa tupla tem o mesmo significado que a duração associada aos lugares no modelo OCPN.

$SYN: T \rightarrow \{\text{or, strong_or, and, weak_and, master, or-master, and-master, weak-master, strong-master}\}$ é a função de tipagem que faz a distinção entre as diferentes regras de disparo, levando a diferentes semânticas de sincronização entre cadeias. A sincronização dentro das cadeias não faz uso de transições tipadas e segue as regras de disparo das redes de Petri convencionais.

A figura 3.12 apresenta, brevemente, a notação gráfica do modelo TSPN.

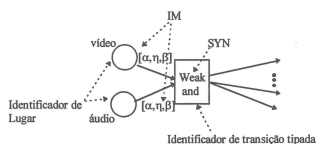


Figura 3.12 - Notação gráfica do modelo TSPN

Em termos dos requisitos de aplicações hipermídia, pode-se dizer que são efetivamente satisfeitos os relativos à estruturação da informação, separação entre estrutura e conteúdo da informação, especificação em vários níveis de granularidade, intervalos de apresentação, atraso e "jitter". Alguns dos demais podem, ainda, ser especificados implicitamente em função de outros elementos da especificação.

3.2.9 O modelo TSPN_{UI}

Uma proposta é apresentada em [Coo95], na qual é possível a especificação de requisitos de sincronização com granularidade fina, incluindo o "jitter", além da especificação de interações com o usuário. O modelo é denominado TSPN_{UI} ("Time Stream Petri Net with User Interaction") e constitui um esforço de integração de características de especificação de sincronização do modelo TSPN com uma versão simplificada, porém mais poderosa, das interações com o usuário permitidas pelo modelo DTPN.

Nesse modelo, como no TSPN, as regras de disparo de transições das redes de Petri padrão são utilizadas para a obtenção de sincronização dentro das cadeias de dados, enquanto que, para a sincronização entre cadeias, essas regras são modificadas. O intervalo de disparo é reavaliado através de um ajuste nos componentes da tupla $[\alpha, \eta, \beta]$ de acordo com a quantidade de tempo pela qual a execução já tiver sido reduzida. Assim, a tupla do $TSPN_{UI}$ recebe um componente adicional e , representando o tempo de execução.

As interações primitivas com o usuário descritas no modelo $TSPN_{UI}$ incluem: “saltar para frente (n unidades de tempo)”, “inverter a direção da apresentação”, “congelar”, “retomar”, “alterar a velocidade (n vezes)” e “iniciar”. Existem duas hierarquias encapsulando essas operações: a “Data, Forward & Reverse Hierarchy” e a “Data Forward (Reverse) Hierarchy”, como mostrado nas figuras 3.13-a e 3.13-b, respectivamente. As hierarquias “forward” e “reverse” são as mesmas, assumindo que as interações com o usuário aplicáveis a uma são também aplicáveis à outra. Essas duas hierarquias fornecem um conjunto flexível de notações para a especificação de interações com usuários, dado que outras interações podem ser construídas através da composição das primitivas. A notação gráfica do modelo é a fornecida pelas hierarquias. Formalmente, o modelo $TSPN_{UI}$ é definido como segue:

$$C_{TSPN_{UI}} = \langle P, T, W_{pt}, W_{tp}, M(0), IM, Rs, SYN, C \rangle \text{ onde}$$

$\langle P, T, W_{pt}, W_{tp}, M(0) \rangle$ define uma rede de Petri marcada ponderada.

$Rs: P \rightarrow (r_1, r_2, \dots, r_n)$ é um mapeamento de recursos para lugares.

$A \subseteq \{T \times P\} \cup \{P \times T\}$ é um conjunto de arcos.

$B = \{b = \langle p, t \rangle \in T \times P \mid W_{tp} \neq 0\}$ é o subconjunto dos arcos que vão das transições para os lugares.

$C \subseteq P \times T; A \cap C = \emptyset$. C representa os arcos de escape.

$IM: B \rightarrow (Q^+ \cup \infty) \times (Q^+ \cup \infty) \times (Q^+ \cup \infty) \times (Q^+ \cup \infty)$ é uma função que mapeia os arcos que entram nas transições para números racionais positivos, representando os valores da tupla $[\alpha, \eta, e, \beta]$. $IM(a_i) = (\alpha_i, \eta_i, e_i, \beta_i)$ é a definição estática inicial.

$SYN: T \rightarrow \{\text{or, strong_or, and, weak_and, master, or_master, and_master, weak_master, strong_master}\}$ é a função de tipagem que faz a distinção entre as diferentes regras de disparo, levando a diferentes semânticas de sincronização entre cadeias.

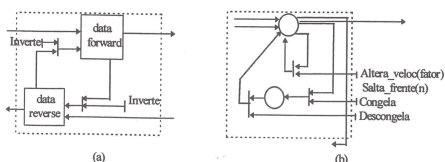


Figura 3.13 - Blocos de construção do modelo $TSPN_{UI}$

O modelo $TSPN_{UI}$ tem uma grande capacidade de especificação; possibilita que a especificação seja construída e lida em vários níveis de granularidade e é relativamente econômico em termos de tamanho do documento de especificação. Entre os modelos baseados em redes de Petri para a especificação de aplicações exclusivamente de multimídia, o $TSPN_{UI}$ parece ser o mais completo.

3.2.10 O modelo MHPN

Em [Wan95] é apresentada outra abordagem baseada em redes de Petri em que há preocupação com a interação com o usuário, como no $TSPN_{UI}$, e com requisitos de aplicações hipermídia, além da especificação de alguns requisitos de sincronização na forma encontrada em modelos como o OCPN. O modelo proposto, chamado MHPN ("Multimedia Hypermedia Petri Net"), estende as redes de Petri através da associação de novos elementos aos componentes da rede. O MHPN busca cobrir as deficiências de outros modelos baseados em redes de Petri como o Trellis, que não se preocupa em especificar sincronização, e o OCPN que, apesar de especificar requisitos temporais (intervalos de apresentação), não se preocupa com entradas do usuário para o serviço multimídia.

As extensões que complementam a rede de Petri no modelo MHPN são objetos MHEG, caracterizando a abordagem adotada como uma tradução da modelagem conceitual em objetos implementacionais já padronizados e passíveis de codificação e apresentação bem conhecidas. Com essa abordagem, o modelo permite que se especifique a estrutura do hiperdocumento de forma independente dos dados, sem maiores preocupações com características de apresentação, satisfazendo requisitos como estruturação da informação, separação entre conteúdo e estrutura da informação e apresentação da informação. A colaboração dos componentes do modelo pode ser rapidamente resumida da seguinte maneira: a rede de Petri fornece a estrutura e a semântica de navegação, tornando-se a máquina de hipermídia subjacente, e os objetos MHEG possibilitam o armazenamento, algumas características adicionais de estruturação e a forma de apresentação do documento.

Um documento MHPN é uma héptupla $H = \langle PN, B, C, E, I, O, A_1 \rangle$ onde

PN é uma Petri Net. $PN = \langle P, T, I \rangle$ (lugares, transições e vértices ou arcos).

B é um conjunto de objetos MHEG básicos

C é um conjunto de objetos MHEG compostos

E é um conjunto de objetos MHEG Conjunto de Ações Condicionais

I é um conjunto de objetos MHEG de entrada

O é um conjunto de canais de saída

A_1 é uma associação lógica para H .

$A_1 = \langle B_1, I_1, H_1, O_1, I_2, C_1, E_1 \rangle$ onde

$$\begin{array}{l}
 B_1: P \rightarrow B U \{v\} \\
 I_1: P \rightarrow I U \{v\} \\
 H_1: P \rightarrow H U \{v\} \\
 O_1: P \rightarrow O U \{v\}
 \end{array}
 \left|
 \begin{array}{l}
 I_2: T \rightarrow I U \{v\} \\
 C_1: V \rightarrow C U \{v\} \\
 E_1: T \rightarrow E U \{v\} \\
 \{v\} \text{ é um valor nulo}
 \end{array}
 \right.$$

As associações acima complementam a rede de Petri no modelo MHPN da seguinte forma: lugares podem ser associados a objetos básicos (associação B_1), objetos de entrada (associação I_1), objetos hiperdocumento, isto é, outros hiperdocumentos (associação H_1), e canais de saída (associação O_1). Transições podem ser associadas a objetos de entrada (associação I_2) e objetos "conjunto de ações condicionais" (associação E_1) e, finalmente, vértices ou arcos podem ser associados a objetos compostos (associação C_1). A figura 3.14 ilustra a extensão realizada pelo método MHPN.

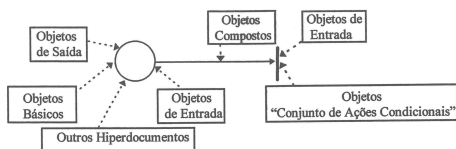


Figura 3.14 - Extensão realizada pelo MHPN sobre as redes de Petri convencionais

Com essas associações é possível estruturar um hiperdocumento, armazená-lo e torná-lo apresentável de forma simples através de ferramentas compatíveis com o padrão MHEG. É importante dizer que o objeto MHEG composto possui uma parte de informação estrutural e outra de dados, a qual não é utilizada nas associações do MHPN. Na parte de informação estrutural existem atributos que possibilitam a especificação de relacionamentos espaço-temporais entre os objetos, cobrindo os requisitos de sincronização e apresentação da informação não tratados pelos modelos Trellis e OCPN. Há de se notar, no entanto, que vários requisitos como os de coordenação espacial da informação, por exemplo, não podem ser satisfeitos em nível conceitual, mas somente na implementação.

3.2.11 O modelo HTSPN

O modelo HTSPN ("Hierarchical Time Stream Petri Net") [Sen96] é uma extensão do modelo TSPN voltada para a especificação de requisitos de aplicações hipermídia. O foco desse modelo, como é o caso dos modelos apresentados anteriormente, está na especificação de documentos e não de aplicações cujas instâncias são hiperdocumentos.

As características das HTSPN que as diferenciam das TSPN são a possibilidade de estruturação através de STSPNs ("Structured Time Stream Petri Nets"), que são TSPNs cujos lugares podem especificar além de um recurso, uma outra TSPN. Com esse mecanismo, a

especificação, tanto estrutural quanto temporal, de um hiperdocumento pode ser realizada em níveis variáveis de abstração e, conseqüentemente, de granularidade. Além do uso de STSPNs como recurso de estruturação, o modelo HTSPN introduz um novo tipo de lugar, chamado de *lugar do tipo ligação*. Esse tipo de lugar modela a presença de uma ligação de hipertexto, a qual pode possuir uma anotação do tipo $[\alpha, \eta, \beta]$, definindo restrições temporais sobre essa ligação. Com isso, além da modelagem da ligação convencional, o modelo suporta também a noção de hipermissão ativa, ou seja, os requisitos relativos à noção de visita a um nó e ligações temporais são satisfeitos.

Segundo Wilrich, em [Wil96], uma HTSPN é definida como uma quintupla $\langle R, C, A, F_C, F_A \rangle$ em que:

R é uma TSPN denominada raiz da especificação hierárquica, estendida para prover as especificações de lugares do tipo ligação, lugares atômicos e lugares compostos (contendo sub-TSPNs). Cada lugar do tipo atômico ou do tipo composto em R é associado, respectivamente pelas funções F_C e F_A , com uma STSPN definida em C ou A . Lugares do tipo ligação e do tipo composto são representados graficamente por círculos em negrito e tracejados, respectivamente.

$C = (C_1, \dots, C_k)$ é uma tupla de STSPNs, com cardinalidade igual à cardinalidade do conjunto de lugares do tipo composto definidos sobre R e C . A função F_C permite a associação entre um lugar do tipo composto definido sobre R ou C com uma STSPN definida em C . Uma restrição imposta sobre esses mapeamentos é a de que uma sub-TSPN não pode conter um lugar do tipo composto associado direta ou indiretamente a ela mesma.

$A = (A_1, \dots, A_k)$ é uma tupla de STSPNs. Cada STSPN em A é relacionada a lugares do tipo atômico em R e C pela função F_A . Essas STSPNs são dedicadas à modelagem de restrições de sincronização intramídia, ou seja, dentro das cadeias de dados. Dessa forma, essas sub-TSPNs são usualmente mais simples e são especificadas em níveis mais finos de granularidade do que aquelas contidas em C .

As regras de disparo do modelo TSPN são modificadas para suportar os novos recursos notacionais e semânticos, de forma que se considera que todas as diferentes sub-redes que compõem uma HTSPN executam como se fossem uma só rede. Assim, pode-se dizer que todas as sub-redes compartilham o mesmo relógio global. O disparo de uma transição t é submetido às restrições usuais de uma TSPN com as seguintes extensões:

Seja o prefixo de uma transição t o conjunto de lugares da rede tal que existem arcos que associam esses lugares a t como lugares de entrada. Seja o sufixo de t o conjunto de lugares da rede tal que existem arcos que associam esse lugares a t como lugares de saída. Quando t dispara ocorre que:

- retiram-se todas as fichas das sub-redes relacionadas a lugares de tipo composto ou de tipo atômico pertencentes ao prefixo de t .
- marcam-se os lugares das sub-redes relacionadas aos lugares do tipo composto ou atômico pertencentes ao sufixo de t .

A figura 3.15 ilustra a notação gráfica do modelo HTSPN.

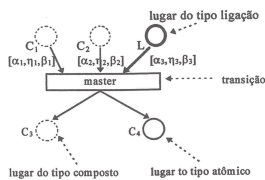


Figura 3.15 - Notação gráfica do modelo HTSPN

No exemplo da figura 3.15, pode-se notar um lugar do tipo ligação L associado a uma transição do tipo "master". Se L for considerado o "master" da transição, então se tem a definição de uma ligação temporal controlada por esse lugar. Dessa forma, a transição irá disparar se a ligação especificada por L for seguida (desde que α_3 unidades de tempo tenham se passado) ou se, após β_3 unidades de tempo desde a habilitação da transição, a ligação não for seguida.

O modelo HTSPN atende à maior parte dos requisitos de aplicações hipermedia apresentados no capítulo 2, destacando-se, entre eles, a especificação em níveis variáveis de abstração e granularidade, estruturação da informação, requisitos relativos à sincronização e ligações temporais.

3.3 Modelos baseados em outras técnicas

Alguns dos modelos apresentados até este ponto permitem a especificação de interações com o usuário e, em alguns casos, de ligações de hipertexto, definindo, de alguma forma, uma semântica de navegação baseada na execução de uma rede de Petri subjacente. Os modelos apresentados nesta seção são baseados em outras abordagens e, por isso, oferecem soluções bastante diversas em termos de notação e de embasamento teórico.

3.3.1 Os modelos HDM e RMDM

Derivado da área de bases de dados, o modelo HDM ("Hypertext Design Model") [Gar93] é um modelo de hipertextos que tem ponto de partida no domínio da aplicação. O modelo propõe uma abordagem baseada em esquemas, como os esquemas de bases de dados, que podem ser posteriormente instanciados com informação específica. Os elementos básicos do modelo são entidades, que são constituídas de componentes, que por sua vez são formados por unidades. As unidades são as porções do modelo que contêm a informação, enquanto que os componentes e classes são mecanismos lógicos de organização da informação. As ligações podem ser utilizadas para relacionar entidades, componentes ou unidades, além de exercerem sua função tradicional de conectar informações de tipos diferentes. O modelo não está relacionado com qualquer semântica de navegação específica, permitindo que o implementador da apresentação especifique a sua própria. Os aspectos temporais relacionados à multimídia não são tratados pelo HDM, de forma que esse modelo só trata dos requisitos relacionados com hipertexto. Por outro lado, o HDM é um modelo cujo foco está voltado para os requisitos

relativos à obtenção e organização da informação, requisito não satisfeito por qualquer um dos modelos apresentados anteriormente.

Similarmente, o modelo RMDM (“Relationship Management Data Model”) [Isa95] procura atender, de forma mais elaborada, aos mesmos requisitos, caracterizando-se, assim como o HDM, como um modelo capaz de especificar aplicações instanciáveis, ou seja, aplicações que apresentam instâncias de esquemas de hiperdocumentos sob demanda, em tempo de navegação. Basicamente, os modelos HDM e RMDM só diferem em detalhes notacionais, mas pode-se dizer que estão baseados nos mesmos conceitos relativos à modelagem de dados. O diferencial entre esses modelos é a utilização do RMDM dentro de um contexto mais amplo: o de uma metodologia para o desenvolvimento de aplicações hipermídia, denominada RMM (“Relationship Management Methodology”), também descrita em [Isa95].

3.3.2 O modelo de referência Dexter para hipertexto

O modelo de referência Dexter para hipertexto [Hal90] tem o objetivo de ser um padrão de construção para sistemas de hipertexto. As abordagens vistas até aqui se referem à especificação de documentos, sejam eles hiperdocumentos, documentos multimídia ou esquemas instanciáveis de documentos. O modelo Dexter refere-se a sistemas de hipertexto, ou seja, softwares que suportam a autoria e a navegação de hiperdocumentos, utilizando algum modelo subjacente. O modelo propõe nós, ligações e âncoras (o efeito visual da ligação) como objetos de baixo nível para a construção dos aspectos estáticos dos sistemas e não modela a estrutura ou o conteúdo da informação dentro das redes de hipertexto, como pode ser visto na figura 3.16. Esses aspectos, bem como seu leiaute e propriedades de apresentação são considerados externos ao modelo. Dessa forma, o modelo Dexter, por si só, não constitui uma abordagem de especificação de hiperdocumentos, mas pode servir como base para o entendimento, derivação e até mesmo para a definição da terminologia de novos modelos.

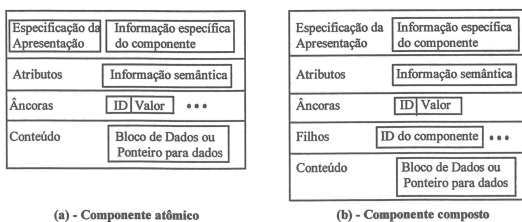


Figura 3.16 - Modelo Dexter

Como se pode ver na figura 3.16, o modelo Dexter sugere que o conteúdo da informação de um hipertexto seja separado da sua estrutura, o que pode ser obtido através da divisão efetiva dos componentes em níveis ou de mapeamentos que tornem a especificação conceitualmente equivalente à proposta pelo modelo.

3.3.3 O "Amsterdam Hypermedia Model"

O AHM ("Amsterdam Hypermedia Model") [Har94] é baseado na estrutura proposta pelo modelo Dexter e reúne características de hipertexto e multimídia, satisfazendo grande parte dos requisitos de aplicações hipermultimídia. Os requisitos multimídia são especificados com base em um modelo de multimídia denominado CMIF ("CWI Multimedia Interchange Format"). O ponto de partida do modelo é uma classificação das abordagens de integração de multimídia e de hipertexto, as quais são descritas brevemente a seguir.

A abordagem de "estrutura escondida" (utilizada na figura 3.3) coloca um objeto multimídia encapsulado em um componente do hipertexto, não modificando a semântica de navegação. O problema com essa abordagem é que a especificação não satisfaz os requisitos referentes à multimídia, já que o objeto multimídia incorporado é considerado pronto para a apresentação e, portanto, especificado através de qualquer outro modelo, eventualmente desconhecido. Essa abordagem pode ser satisfatória para um certo número de aplicações, mas certamente será insuficiente, por exemplo, para casos em que a aplicação seja distribuída.

O problema é atacado de forma contrária pela abordagem de "estrutura separada". Nessa abordagem os elementos de cada mídia são considerados componentes e são utilizadas ligações multi-destino para que todos os componentes sejam ativados. A dificuldade com a especificação temporal e o tamanho da especificação tornam essa abordagem inviável, dado que em aplicações genéricas não se espera que a apresentação de todos os componentes se inicie ao mesmo tempo, de forma que a alternativa para solucionar o problema da especificação temporal seria acrescentar essa informação à estrutura da ligação, o que sobrecarregaria a tarefa de autoria.

Uma solução intermediária é a abordagem de "estrutura composta" em que alguns elementos multimídia são encapsulados por componentes do hipertexto. O que ocorre com essa abordagem é que há uma modularização da solução, ou seja, os problemas de sincronização são resolvidos em porções menores e a especificação se torna mais legível. O problema é que, como na abordagem da "estrutura separada", os problemas de especificação temporal persistem, agora dentro de cada componente.

O modelo Amsterdã busca um compromisso entre essas abordagens, utilizando informação temporal associada aos componentes atômicos e compostos do modelo Dexter, além de modificar a definição do componente composto, não mais permitindo a sua associação direta com dados, mas somente a existência de referências a componentes atômicos, os quais possuem associação com dados.

A sincronização é determinada pela forma como o componente composto referencia os componentes atômicos associados a ele. Essas formas podem ser "choice" ou "parallel", ou seja, as cadeias de dados dos componentes atômicos podem ser apresentadas exclusiva ou concorrentemente. A informação referente às relações temporais entre as cadeias mostradas em paralelo é obtida dos dados temporais dos componentes atômicos. Para tratar da sincronização entre cadeias, o modelo lança mão de arcos de sincronização. Esses arcos são usados para indicar se duas cadeias devem sincronizar suas apresentações em algum ponto, permitindo a especificação de tipos de sincronização e de "jitter". Não parece haver tratamentos diferenciados para a especificação de sincronização entre cadeias de dados ou dentro das cadeias, a não ser pelo uso de arcos de sincronização.

Outra característica do modelo é a disposição dos diagramas em uma linha de tempo, com blocos representando as cadeias de dados e arcos direcionados representando ligações. O modelo trata ainda de questões relacionadas com o comportamento da apresentação multimídia quando uma ligação é seguida. Esse requisito é especificado através de contextos de "link" (ou contextos de informação), que são representados por componentes compostos que encapsulam os componentes envolvidos na origem e no destino de uma ligação, permitindo especificar se alguns dados irão persistir ou se suas apresentações serão encerradas.

O AHM utiliza o conceito de canais, dispositivos lógicos que representam os meios físicos pelos quais a informação deve ser apresentada. Com esse recurso, o modelo permite especificar atributos de apresentação de alto nível. O volume do áudio, velocidade da apresentação e preferências de cor são exemplos de atributos de apresentação de alto nível controlados pelos canais no modelo AHM. O uso de canais permite que a mesma apresentação possa ser configurada de acordo com grupos específicos de usuários. Por exemplo, em uma apresentação em que há vídeo, áudio e legendas (texto), pode-se utilizar áudio em inglês e vários canais de texto com legendas em várias línguas. Os canais de texto seriam ativados e desativados de acordo com a língua falada pelo usuário, em tempo de apresentação. A figura 3.17 apresenta a notação gráfica do modelo Amsterdã.

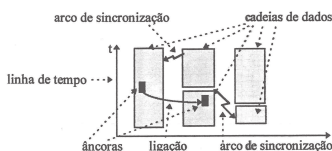


Figura 3.17 - Notação gráfica do modelo Amsterdã de hipermídia

Em termos dos requisitos de aplicações hipermídia, pode-se dizer que o modelo Amsterdã não permite especificar aqueles relativos à obtenção e organização da informação, além de locais de origem e destino, taxas de "throughput", sensibilidade a erros (sincronização) e coordenação espacial (apresentação), requisito que pode, no entanto, ser especificado através dos canais.

3.4 Statecharts e hipermídia

Esta seção é dedicada ao HMBS ("Hypertext Model Based on Statecharts") [Oli95], o modelo que constitui o ponto de partida para as propostas apresentadas nos capítulos subsequentes deste trabalho. A principal característica desse modelo é a utilização de statecharts [Har87a, Har87b] como técnica subjacente, de maneira similar ao uso de redes de Petri pelo modelo Trellis, apresentado na seção 3.2.5. Dessa forma, para um melhor entendimento, antes da apresentação do modelo, é apresentada uma introdução à linguagem de statecharts, a qual pode ser complementada pelas informações contidas em [Har87a e Har87b].

A definição formal do modelo TSPN é a que segue:

$C_{TSPN} = \langle P, T, B, F, M, IM, SYN \rangle$ onde

$\langle P, T, B, F, M \rangle$ define uma rede de Petri marcada ponderada, ou seja, uma rede de Petri onde o peso atribuído a um arco corresponde ao número de fichas que ele transporta.

se $A = \{a = \langle p, t \rangle \in P \times T \mid (B(\langle p, t \rangle) \neq 0)\}$ é o conjunto de arcos saindo dos lugares, então:

$IM: A \rightarrow Q^+ \times Q^+ \times (Q^+ \cup \infty)$ é o mapeamento de arcos que entram nas transições para números racionais positivos representando os valores da tupla $[\alpha, \eta, \beta]$. A duração representada por η nessa tupla tem o mesmo significado que a duração associada aos lugares no modelo OCPN.

$SYN: T \rightarrow \{\text{or, strong_or, and, weak_and, master, or-master, and-master, weak-master, strong-master}\}$ é a função de tipagem que faz a distinção entre as diferentes regras de disparo, levando a diferentes semânticas de sincronização entre cadeias. A sincronização dentro das cadeias não faz uso de transições tipadas e segue as regras de disparo das redes de Petri convencionais.

A figura 3.12 apresenta, brevemente, a notação gráfica do modelo TSPN.

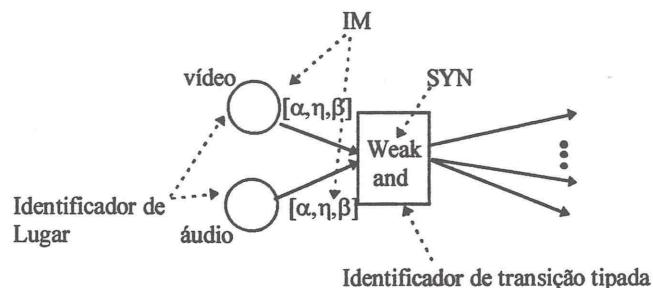


Figura 3.12 - Notação gráfica do modelo TSPN

Em termos dos requisitos de aplicações hipermídia, pode-se dizer que são efetivamente satisfeitos os relativos à estruturação da informação, separação entre estrutura e conteúdo da informação, especificação em vários níveis de granularidade, intervalos de apresentação, atraso e “jitter”. Alguns dos demais podem, ainda, ser especificados implicitamente em função de outros elementos da especificação.

3.2.9 O modelo TSPN_{UI}

Uma proposta é apresentada em [Coo95], na qual é possível a especificação de requisitos de sincronização com granularidade fina, incluindo o “jitter”, além da especificação de interações com o usuário. O modelo é denominado TSPN_{UI} (“Time Stream Petri Net with User Interaction”) e constitui um esforço de integração de características de especificação de sincronização do modelo TSPN com uma versão simplificada, porém mais poderosa, das interações com o usuário permitidas pelo modelo DTPN.

Um statechart possui um conjunto de expressões que, de forma simples, contém variáveis (que são sempre de escopo global), números e expressões algébricas envolvendo esses elementos.

O conjunto de condições um statechart inclui, como condições primitivas, os valores lógicos *true* e *false*. Além disso, sentenças relacionais envolvendo expressões e sentenças lógicas envolvendo outras condições também são consideradas condições.

Os eventos primitivos de um statechart são aqueles eventos que ocorrem no ambiente do sistema sob especificação. Além desses eventos, os statecharts permitem que um evento nulo (λ) seja representado, correspondendo a um evento que é gerado a cada passo de execução ou simulação do sistema. Expressões do tipo *true* (condição) e *false* (condição) denotam eventos que ocorrem quando o valor de uma condição se torna, respectivamente, verdadeiro ou falso. A expressão *changed* (expressão) denota um evento que ocorre quando o valor uma expressão é modificado. As expressões *exit* (estado) e *entered* (estado) denotam eventos que ocorrem, respectivamente, quando o sistema sai de um estado ou entra em um estado. Combinações de quaisquer das expressões de eventos através de operadores lógicos são consideradas eventos e, adicionalmente, expressões do tipo “evento [condição]” são igualmente consideradas eventos. Finalmente, a expressão *time-out* (evento observado, número de passos) denota um evento que ocorre quando um determinado número de passos decorre após a ocorrência de um evento observado durante a execução do sistema.

As ações de um statechart podem ser a ação nula (μ), uma expressão de atribuição de valor a uma variável ou condição, na forma “a := b”, ou um conjunto de ações separadas por “;”, na forma “a₁;...; a_n”.

As transições representam as possibilidades de alteração no estado do sistema devido à ocorrência de algum evento, seja ele externo (gerado pelo ambiente) ou interno (gerado pela própria aplicação). Uma transição sempre envolve um conjunto de estados de origem, um conjunto de estados de destino e um rótulo. O conjunto de rótulos de transições de um statechart é formado por elementos denotados por *e/a*, relativos aos pares formados por expressões de eventos e de ações. Informalmente, se *e/a* é um rótulo de uma transição *t*, então *t* é disparada pela ocorrência do evento *e*, causando a execução da ação *a*. Pode-se dizer, também informalmente, que a forma geral do rótulo de uma transição é o seguinte: *ev [cond]/a*, onde *ev* é uma expressão de eventos, *cond* é uma expressão condicional aplicada sobre a transição *t* e *a* é um conjunto de ações.

Informalmente, diz-se que uma transição *t* com origem em um conjunto de estados *X*, destino em um conjunto de estados *Y* e com um rótulo “*e [c] / a*”, disparará se o sistema estiver em *X* e *e* ocorrer, desde que a condição *c* seja satisfeita. Dessa forma, as ações em *a* serão executadas e o sistema passará a estar nos estados determinados pelo conjunto *Y*. As transições são representadas por arcos direcionados que ligam os estados de origem aos de destino. Quando mais do que um estado está envolvido no destino ou na origem de uma transição, as extremidades do arco que a representa são ramificadas.

Um símbolo semelhante ao que representa uma transição é o indicador de estado “default”, também chamado de transição “default”. Se, por exemplo, um estado *A*₁ é do tipo OR e é decomposto nos subestados *C* e *D*, de tal forma que o estado *C* possui o indicador de estado “default”, então, havendo uma transição que determina a ativação de *A*₁, o subestado *C*

será ativado. Em termos genéricos, sempre que um estado X do tipo OR for ativado, o subestado de X a ser ativado será aquele ao qual o indicador de estado "default" estiver associado.

Esses conceitos possibilitam a descrição do comportamento de um sistema através de um statechart. Na figura 3.19 é apresentado um exemplo de statechart em que é possível reconhecer suas principais características. O exemplo de estado "default" citado pode ser verificado na figura.

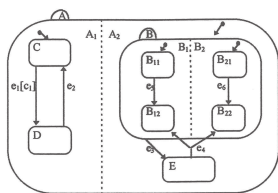


Figura 3.19 Um exemplo de statechart

A semântica de execução dos statecharts é baseada em um modelo discreto de tempo, segundo o qual os estados de um sistema são duráveis, enquanto que as transições são instantâneas. O termo passo de execução ou de simulação tem origem nesse modelo de tempo. Diz-se que em um passo de execução o statechart assume uma configuração de estados, a qual corresponde ao conjunto dos estados básicos ativos naquele passo. Dessa forma, a cada passo, obtém-se uma nova configuração do statechart, a qual pode ou não ser igual à configuração anterior, dependendo de quais transições dispararem.

A configuração inicial de um statechart é aquela obtida pela ativação de seu estado raiz e, recursivamente, de todos os subestados "default". A figura 3.20 apresenta a configuração inicial para o statechart apresentado na figura 3.19.

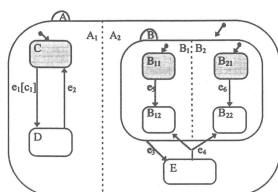


Figura 3.20 Configuração inicial

No exemplo das figuras 3.19 e 3.20, pode-se verificar a presença de duas das três principais características de statecharts: concorrência entre estados ($A_1 \perp A_2$, $B_1 \perp B_2$) e composição hierárquica de estados. No entanto, um dos mecanismos mais poderosos da técnica é o de difusão de eventos, através da geração de eventos internos. Em termos notacionais, a geração de um evento interno é obtida através da especificação de um evento primitivo na expressão de ações contida no rótulo de uma transição. Dessa forma, quando a transição disparar, o evento especificado em sua expressão de ações será gerado e sentido por todos os componentes concorrentes que estiverem ativos no passo, os quais, se for o caso, poderão reagir e evoluir.

A figura 3.21 é uma modificação da figura 3.20 que apresenta a nova configuração do statechart (a partir da configuração inicial e com a ocorrência de e_3) caso a transição que parte do estado B_{11} e chega ao estado B_{21} tenha seu rótulo modificado para " e_3 / e_6 ", significando, portanto, que quando essa transição disparar, o evento e_6 será gerado.

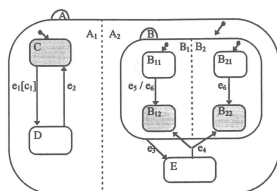


Figura 3.21 - Exemplo de disparo de transição com difusão de evento interno

Note-se que se o rótulo da transição permanecesse " e_3 ", a componente OR concorrente B_2 não evoluiria e o seu subestado ativo continuaria sendo B_{21} . Outra característica decorrente da estruturação hierárquica dos estados é a possibilidade de se especificar transições que, mesmo partindo de um único estado de origem, podem desativar (ou ativar) conjuntos de estados. Esse é o caso da transição associada ao evento e_3 nas figuras 3.19 a 3.21, que parte do estado B e chega ao estado E . A figura 3.22 apresenta uma nova configuração a partir daquela apresentada na figura 3.21 se o evento e_3 ocorrer.

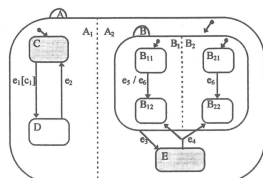


Figura 3.22 - Configuração após o a ocorrência de e_3

Outro recurso importante em statecharts é o mecanismo de história. Quando o sistema entra em um estado X do tipo OR com símbolo de história associado, a configuração relativa a X torna-se igual à última configuração encontrada quando o sistema esteve em X pela última vez, o que pode, potencialmente, diferir da configuração determinada pelos indicadores de estados "default". Somente quando se tratar da primeira entrada do sistema no estado X , a configuração será determinada pelos indicadores de estados "default".

Finalmente, com o objetivo de se criar um elo entre a especificação comportamental e a especificação funcional de um sistema em desenvolvimento, permite-se que statecharts possuam atividades, que são funções do sistema, associadas aos seus estados. Basicamente, permite-se que uma atividade seja ativada quando o sistema entrar em um estado ("On entry"), quando o sistema sair de um estado ("On exit"), ou enquanto estiver em um estado ("throughout"). Dessa forma, o statechart pode constituir o elemento controlador das funções de um sistema sob especificação.

O conjunto de notações apresentado nesta seção apresenta a técnica de statecharts de forma rápida e objetiva, sem a intenção de ser uma introdução definitiva à linguagem. Certamente, outros recursos de notação e outras formas de utilização dos recursos aqui apresentados podem ser explorados, aumentando o poder de especificação da técnica.

3.4.2 Statecharts temporais

Os statecharts foram estendidos para uma versão voltada especificamente à especificação de sistemas de tempo real, dando origem aos "Timed Statecharts" [Pnu92]. Nessa extensão, as expressões de eventos são acrescidas de um intervalo $[l, u]$ que determina as fronteiras temporais mínima e máxima do disparo da transição. Os parâmetros temporais associados à transição determinam o tempo durante o qual o sistema permanecerá no conjunto de estados de origem, bem como o atraso permitido para que a transição dispare após a fronteira mínima ter sido alcançada. Em "Timed Statecharts" não são permitidas transições entre níveis nem referências a nomes de estados. Dessa forma condições mais avançadas de statecharts como " $m(s)$ ", que testa se um estado s está ativo, não são definidas.

Diferentemente de statecharts, "timed statecharts" (ou statecharts temporais) são assíncronos, com concorrência sendo modelada como intercalação. Os únicos passos que são síncronos nessa técnica são aqueles em que o relógio global do sistema é incrementado. O disparo da transição é instantâneo, de forma que em qualquer instante é possível saber qual é o estado do sistema. A figura 3.23 apresenta um exemplo de statechart temporal.

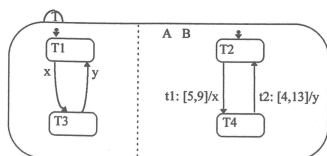


Figura 3.23 - Exemplo de statechart temporal

A definição formal do modelo TSPN é a que segue:

$C_{TSPN} = \langle P, T, B, F, M, IM, SYN \rangle$ onde

$\langle P, T, B, F, M \rangle$ define uma rede de Petri marcada ponderada, ou seja, uma rede de Petri onde o peso atribuído a um arco corresponde ao número de fichas que ele transporta.

se $A = \{a = \langle p, t \rangle \in P \times T \mid (B(\langle p, t \rangle) \neq 0)\}$ é o conjunto de arcos saindo dos lugares, então:

$IM: A \rightarrow Q^+ \times Q^+ \times (Q^+ \cup \infty)$ é o mapeamento de arcos que entram nas transições para números racionais positivos representando os valores da tupla $[\alpha, \eta, \beta]$. A duração representada por η nessa tupla tem o mesmo significado que a duração associada aos lugares no modelo OCPN.

$SYN: T \rightarrow \{\text{or, strong_or, and, weak_and, master, or-master, and-master, weak-master, strong-master}\}$ é a função de tipagem que faz a distinção entre as diferentes regras de disparo, levando a diferentes semânticas de sincronização entre cadeias. A sincronização dentro das cadeias não faz uso de transições tipadas e segue as regras de disparo das redes de Petri convencionais.

A figura 3.12 apresenta, brevemente, a notação gráfica do modelo TSPN.

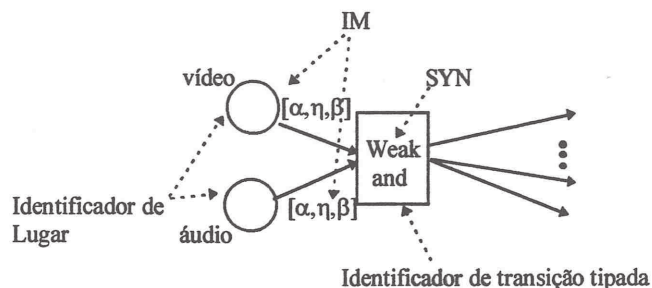


Figura 3.12 - Notação gráfica do modelo TSPN

Em termos dos requisitos de aplicações hipermídia, pode-se dizer que são efetivamente satisfeitos os relativos à estruturação da informação, separação entre estrutura e conteúdo da informação, especificação em vários níveis de granularidade, intervalos de apresentação, atraso e “jitter”. Alguns dos demais podem, ainda, ser especificados implicitamente em função de outros elementos da especificação.

3.2.9 O modelo TSPN_{UI}

Uma proposta é apresentada em [Coo95], na qual é possível a especificação de requisitos de sincronização com granularidade fina, incluindo o “jitter”, além da especificação de interações com o usuário. O modelo é denominado TSPN_{UI} (“Time Stream Petri Net with User Interaction”) e constitui um esforço de integração de características de especificação de sincronização do modelo TSPN com uma versão simplificada, porém mais poderosa, das interações com o usuário permitidas pelo modelo DTPN.

- $M: S_i \rightarrow P$ é uma função valorada que mapeia um subconjunto de estados em páginas, onde o conjunto S_i é formado pelos estados do tipo OR e pelos estados básicos do statechart.
- L é o nível de “browsing” do hiperdocumento, ou nível de visibilidade, usado para definir a profundidade da hierarquia do statechart a ser apresentada durante a navegação.
- $V: P \rightarrow R$ é o relacionamento de visualização que associa cada página do hiperdocumento com um único “reader” capaz de interpretá-la.

A definição de statechart utilizada pelo HMBS é um subconjunto da definição apresentada em [Har87a], possuindo somente as características necessárias à especificação de hipertextos. Ela não inclui o mecanismo de propagação de eventos através de ações, nem os símbolos de história, além de outras características mais complexas que a técnica oferece. Dessa forma, um statechart ST no HMBS é definido como uma óptupla $ST = \langle S, \rho, \psi, \delta, V, C, E, T \rangle$ onde:

- $S = \{s_1, s_2, \dots, s_n\}$ é o conjunto de estados.
- $\rho: S \rightarrow 2^S$ é a função de hierarquia que associa cada estado ao conjunto de seus subestados.
- $\psi: S \rightarrow \{\text{AND}, \text{OR}\}$ é a função de tipo de decomposição.
- $\delta: S \rightarrow 2^S$ é a função “default”.
- V : é o conjunto de expressões contendo nomes de variáveis de um conjunto de variáveis lógicas V_F com valores iniciais conhecidos. O conjunto $V = V_T \cup V_F$ é, portanto, dado por $V_T \cup V_F$, em que V_T é o conjunto de variáveis cujo valor inicial é *true* e V_F é o conjunto de variáveis cujo valor inicial é *false*.
- C é o conjunto de condições, que podem ser T (*true*), F (*false*) ou expressões lógicas.
- E é o conjunto de expressões de eventos, que podem estar contidas no conjunto de eventos primitivos E_p , ou podem estar combinadas com condições, na forma $e [c]$.
- T é o conjunto de transições, definidas por triplas (X, r, Y) , onde X é o conjunto de estados de origem, r é o rótulo (uma expressão de eventos) e Y é o conjunto de estados de destino. Uma transição dispara se a expressão de eventos r resultar em T (*true*) e se o sistema estiver no conjunto de estados X .

O HMBS permite que o autor defina o nível de visualização do hiperdocumento através do atributo L (“level”). Um hiperdocumento para o qual $L = 0$ tem exibidas todas as páginas associadas aos estados básicos que estiverem ativos em uma determinada configuração. Com $L = 1$, as páginas exibidas serão aquelas pertencentes ao conjunto das páginas mapeadas para os estados pertencentes à configuração básica e aos seus ancestrais imediatos, e assim por diante para $L > 1$.

Explorando a hierarquia de estados do statechart, o HMBS permite que sejam fornecidas visões hierárquicas do hiperdocumento. Trata-se de um mecanismo de navegação que permite que o usuário visualize as páginas associadas a estados em níveis de hierarquia acima daquele em que está navegando, diminuindo a sua eventual desorientação. Um exemplo

disso é a navegação por um hiperdocumento que contém um livro. Com o mecanismo de visões hierárquicas, disponibilizado pelo HMBS pela operação "Show-hview", é possível ao usuário subir na hierarquia para descobrir em que seção, capítulo, etc, ele está navegando em um determinado momento.

A capacidade de especificar concorrência entre estados dos statecharts possibilita ao HMBS especificar a exibição simultânea de páginas. Pelo mesmo motivo, o modelo permite a definição de caminhos concorrentes de navegação, ou seja, podem ser definidas ligações independentes, as quais podem causar a modificação de apenas uma parte da apresentação. Essas características enriquecem o modelo e o tornam bastante flexível para a especificação de vários tipos de hiperdocumentos.

Pode-se dizer, portanto, que o HMBS é um modelo abrangente para o domínio de hipertextos. No entanto, como será discutido adiante, o modelo não é apropriado para a especificação de aplicações do domínio de hiperídia, uma vez que vários requisitos desse tipo de aplicação não são satisfeitos.

3.4.4 O modelo HMBS frente aos requisitos de aplicações hiperídia

Esta subseção apresenta uma revisão mais detalhada do modelo HMBS em termos dos requisitos de aplicações hiperídia considerados neste trabalho. Os resultados obtidos apontam as deficiências do modelo HMBS, o que permitiu, para este trabalho, a definição de diretrizes para a criação do modelo XHMBS. Como forma de facilitar o acesso às informações, os resultados são apresentados em forma tabular no final da seção.

3.4.4.1. Requisitos relativos à obtenção e organização da informação

Modelagem do domínio da aplicação

O requisito de modelagem do domínio da aplicação, visto como a capacidade que um modelo tem de apoiar a modelagem semântica ou lógica da aplicação, não é satisfeito pelo modelo HMBS. No entanto, é possível que, entendidas as características semânticas do problema, o modelo seja utilizado com o objetivo de analisar a aplicação.

A abordagem convencional do HMBS, porém, o caracteriza como um modelo de projeto de hiperdocumentos. Assim, presume-se que alguma outra forma de obtenção de informações seja utilizada antes da definição das páginas e até mesmo da estrutura básica do hiperdocumento. O modelo HMBS é considerado um modelo comportamental (ou navegacional) de hiperdocumentos.

Derivação de elementos da aplicação

Esse requisito está associado com o processo de análise da aplicação, ou seja, a fase do desenvolvimento em que são identificados os componentes do hiperdocumento ou dos seus respectivos esquemas. Como dito antes, o modelo HMBS aborda a modelagem em nível de projeto, não possuindo dispositivos sofisticados de análise, como é o caso de mecanismos de derivação de elementos. Assim, esse requisito não pode ser satisfeito pelo modelo.

Estruturação da informação

O modelo HMBS permite que a informação seja estruturada através das três formas descritas na seção 2.2.3. A figura 3.24 apresenta as representações de organização hierárquica, linear e em rede com a notação do modelo. Adicionalmente, devido à definição da semântica de navegação do modelo, baseada na execução do statechart, o fato de se organizar a informação de determinada forma implica em se determinar a maneira segundo a qual essa informação será apresentada.

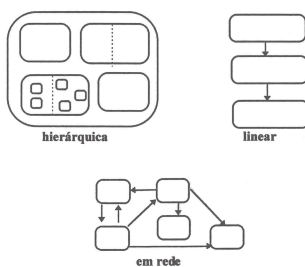


Figura 3.24 - Estruturas básicas de informação em HMBS

Separação entre estrutura e conteúdo da informação

A separação entre estrutura e conteúdo da informação é obtida parcialmente através dos mapeamentos entre estados e páginas. As âncoras são obtidas dos eventos que compõem o rótulo de uma transição. Assim, o autor deve se preocupar com o conteúdo de uma âncora enquanto estiver definindo a estrutura do hiperdocumento, o que é indesejável. Dessa forma, esse requisito é apenas parcialmente satisfeito.

Várias visões da mesma informação

Esse requisito pode ser satisfeito de forma pouco conveniente pelo modelo HMBS. O mapeamento M entre estados e páginas pressupõe que um estado esteja associado a uma única página. Com essa característica, se forem necessárias n visões de toda a informação contida em um hiperdocumento, serão necessárias também n versões de cada página de informação. Isso quer dizer que, se todas as visões forem necessárias em tempo de navegação, deverá haver, além da estrutura normal do hiperdocumento, $n-1$ réplicas organizadas de forma exclusiva, mapeadas para as outras versões das páginas. O usuário poderia, então, selecionar as versões desejadas seguindo as ligações apropriadas.

Versões “sob medida” do mesmo hiperdocumento

O modelo HMBS permite algumas formas de personalização de um mesmo hiperdocumento. A primeira delas é o valor do atributo “Level”, ou nível de visibilidade do hiperdocumento. É possível privilegiar usuários com valores diferentes do nível de visibilidade, de forma que a visão da hierarquia do hiperdocumento se torne mais clara. Além disso, com o uso de variáveis, composição hierárquica de estados e condições associadas às transições, pode-se determinar estruturas de controle de acesso a certas regiões do hiperdocumento. Um usuário poderá receber o conjunto de variáveis com os valores arranjados de forma que seja possível ou não acessar determinadas informações.

Especificação em vários níveis de abstração e de granularidade

Esses requisitos são satisfeitos pelo HMBS através da possibilidade de decomposição de estados do statechart. Um desenvolvedor poderá abordar o processo de autoria em vários níveis de abstração e granularidade graças a essa característica. Normalmente, os detalhes da especificação do hiperdocumento irão aparecer no nível mais baixo de abstração, no qual o nível de granularidade é mais fino. Com o uso de vários níveis de abstração, a especificação do controle de sincronização entre cadeias pode ser realizada em níveis variáveis de granularidade.

Essa característica pode ser melhorada através de abordagens de especificação como o uso de blocos de construção (“building blocks”), que são construções pré-definidas que especificam um determinado comportamento bem conhecido no domínio da aplicação. Essa abordagem possibilita que as especificações tenham uma aparência padronizada e sejam mais concisas, o que as torna mais legíveis.

3.4.4.2 Requisitos relativos à sincronização da apresentação

Locais de origem e de destino

O modelo não permite a especificação dos locais de origem e destino das cadeias de dados. A satisfação desse requisito é deixada por conta do implementador devido à abordagem de especificação do modelo, ou seja, é possível alocar a origem e o destino das cadeias de dados livremente quando a especificação for interpretada ou implementada. Intuitivamente, imagina-se que o local de destino de um hiperdocumento seja a máquina que executa a especificação, mas essa abordagem pode limitar as características de distribuição da aplicação.

Intervalos de apresentação, atraso e “jitter”

Como os statecharts não possuem notação temporal razoável, esses requisitos não podem ser satisfeitos. A única notação possível para representar relações temporais com statecharts é a de “time-outs”, com a qual somente as durações de ativação dos estados poderiam ser especificadas. No entanto, nem mesmo essa notação consta da definição de statecharts utilizada pelo modelo HMBS.

Tipos de fonte, dependência temporal das cadeias de dados, taxas de “throughput” e sensibilidade a erros

Não há mecanismos no modelo HMBS que permitam especificar esses requisitos de forma sistemática. Assim, se o autor da aplicação desejar, podem ser feitas anotações informais na especificação sem que exista, porém, alguma forma de checar se essa especificação é consistente.

3.4.4.3 Requisitos relativos à interação entre usuário e computador

O modelo HMBS não permite que sejam especificadas muitas das operações de controle comuns em apresentações multimídia. Algumas interações, como “iniciar” e “reiniciar”, podem ser representadas como ligações entre componentes. Essa abordagem é bastante restrita, pois, muitas vezes, deseja-se especificar interações que não necessariamente correspondam a ligações, o que poderia levar à desorientação do usuário. Esse conjunto de requisitos pode ser, portanto, apenas parcialmente especificado.

3.4.4.4 Requisitos relativos à apresentação da informação

Coordenação espacial da informação e controle de atributos globais da apresentação

O HMBS não possui qualquer mecanismo que permita especificar a disposição da informação nos dispositivos de apresentação ou o controle sobre os atributos globais da apresentação do hiperdocumento. Presume-se que os “readers” sejam capazes de conter essa informação, mas o modelo não define qualquer mecanismo que permita que o autor especifique esses requisitos.

Tipo de mídia da cadeia de dados

Esse requisito é satisfeito implicitamente através do mapeamento V , que relaciona páginas a “readers”. Essa associação faz com que o tipo de “reader” escolhido determine, ou indique o tipo dos dados contidos em uma cadeia de dados. Essa abordagem não permite que sejam feitas checagens em tempo de autoria, uma vez que, se uma associação equivocada for feita, o erro será percebido somente quando o hiperdocumento for implementado, quando, provavelmente, ocorrerá um erro indicando que o dispositivo de apresentação que não é capaz de apresentar os dados.

3.4.4.5 Requisitos relativos à navegação do hiperdocumento

Noção de visita a um nó

A noção mais elementar da semântica de navegação do HMBS é a de visita. O uso de transições permite que o tempo de apresentação da informação de uma página seja determinado somente pela decisão do usuário de seguir uma ligação. Assim, o conceito convencional de navegação em hipertextos é satisfeito. Como é apresentado adiante, no entanto, o modelo não permite a especificação de ligações temporais e, dessa forma, a noção de visita a um nó determinada temporalmente não é suportada.

Estruturas alternativas de acesso à informação

O modelo permite que a navegação seja feita através da forma tradicional, seguindo as ligações disponíveis, ou através de índices que devem ser disponibilizados pelo sistema de navegação. A disponibilização de índices pelo sistema de navegação é apoiada pela definição de página feita pelo modelo. Apesar de ser feita informalmente, a definição de página diz que uma página é composta de conteúdo e título. Os títulos são utilizados como chaves de busca para compor o índice.

Ligações temporais

Visto que não há notação temporal disponível no modelo, esse requisito não pode ser especificado. As únicas ligações permitidas em um documento HMBS são estáticas, como as encontradas em hipertextos convencionais.

Ligações dinâmicas

O modelo HMBS não possui qualquer forma de especificação de ligações dinâmicas, mesmo porque, essas ligações estão muito mais associadas com o sistema de navegação do que com o próprio modelo. No entanto, os elementos do modelo, como os títulos de páginas e as semânticas de obtenção de contextos (que recuperam os componentes concorrentes do hiperdocumento quando uma página é encontrada através do índice), permitem que mecanismos de busca criem ligações dinâmicas e índices, levando o usuário a informações específicas de forma não-convencional.

Ligações dentro de componentes

Apesar de statecharts permitirem que transições possuam conjuntos de estados de origem iguais aos conjuntos de estados de destino, a semântica de uma transição desse tipo no HMBS é equivalente a uma ação de reinício da apresentação de um determinado contexto de informação. O que se pretende com ligações dentro de componentes é que, para mídias dinâmicas, haja um deslocamento da referência temporal e, assim, a informação que estiver sendo apresentada passe para um ponto diferente daquele que estava sendo apresentado quando da ocorrência da interação. Isso só é possível através de mecanismos de manipulação da referência temporal local de um estado, os quais não estão disponíveis em statecharts.

Contextos de informação

O uso de superestados dos statecharts permite a definição de contextos de informação de hiperdocumentos de forma bastante simples. Um estado-pai determina o contexto em que estão inseridos os seus subestados. Assim, ligações que partem desse estado-pai fazem com que os seus subestados deixem de ser apresentados e ligações que chegam a um estado-pai fazem com que alguns de seus subestados passem a ser apresentados, definindo os contextos de ativação e desativação de uma ligação.

Visão da hierarquia do hiperdocumento

Um ponto forte da especificação baseada em statecharts provida pelo HMBS é a facilidade em se especificar a visualização da hierarquia do hiperdocumento. Esse requisito é

satisfeito pelo HMBS através da combinação da organização hierárquica da informação, do nível de visibilidade do hiperdocumento e das operações de visualização hierárquica ("Show-hview").

3.4.4.6 Resumo da análise do HMBS

Como referência rápida à revisão do modelo HMBS, é apresentada nesta subseção uma tabela (tabela 3.1) que contém os resultados obtidos em uma forma resumida e objetiva. Um trabalho completo de extensão desse modelo visando a satisfação de todos os recursos aqui apresentados deveria completar essa tabela com as formas de especificar os requisitos que constam como não satisfeitos.

Requisito	Satisfeito pelo HMBS	Observação
<i>Modelagem do domínio da aplicação</i>	√	Statecharts podem ser utilizados para análise, mas o HMBS se presta mais especificamente ao projeto de hiperdocumentos. Esse requisito é parcialmente satisfeito.
<i>Derivação de elementos da aplicação</i>	×	Não há mecanismos específicos para análise disponíveis no modelo.
<i>Estruturação da Informação</i>	√	Statecharts permitem organizar os estados de forma linear, hierárquica ou em rede. A organização implica na definição do comportamento da apresentação.
<i>Separação entre estrutura e informação</i>	√	Parcial. As páginas são separadas da estrutura via função <i>M</i> . No entanto, as âncoras têm seus conteúdos especificados pelos rótulos das transições do statechart.
<i>Várias visões da mesma informação</i>	√	Pode ser satisfeito através da replicação de estados e páginas relacionados a cada versão da informação. Abordagem pouco conveniente: demanda replicação da estrutura.
<i>Versões "sob medida" do mesmo hiperdocumento</i>	√	Uso de variáveis e composição hierárquica de estados.
<i>Especificação em vários níveis de abstração</i>	√	Statecharts permitem decomposição.
<i>Especificação em vários níveis de granularidade</i>	√	Statecharts permitem decomposição. Com mais de um nível de abstração, a granularidade pode variar.
<i>Local de origem da cadeia de dados</i>	×	Não há qualquer mecanismo que permita especificar esse requisito.
<i>Local de destino da cadeia de dados</i>	×	O mesmo ocorre com locais de destino. Pode-se presumir que o local de destino é a máquina que executa a especificação, o que diminui a flexibilidade da especificação.

Requisito	Satisfeito pelo HMBS	Observação
<i>Intervalos de apresentação</i>	×	Não há notação temporal no modelo.
<i>Tipos de fonte</i>	×	Não há mecanismos para especificar esse requisito.
<i>Dependência temporal</i>	×	idem.
<i>Atraso</i>	×	Não há notação temporal no modelo.
<i>"Jitter"</i>	×	idem.
<i>Taxas de "throughput"</i>	×	Não há mecanismos para especificar esse requisito.
<i>Sensibilidade a erros</i>	×	idem.
<i>Interação Usuário-Computador</i>	√	Parcial. Todas as interações que podem ser definidas são tratadas como ligações de hipertexto. Interações de multimídia envolvendo tempo não são suportadas.
<i>Coordenação espacial</i>	×	Não há mecanismos para especificar esse requisito.
<i>Controle de atributos globais da apresentação</i>	×	idem
<i>Tipo de meio da cadeia de dados</i>	√	A associação de "readers" a páginas via mapeamento V especifica esse requisito implicitamente.
<i>Noção de visita a um nó</i>	√	A semântica de navegação é baseada na noção de visita às páginas. O usuário determina o final da visita quando segue uma ligação. Hipermídia ativa não é suportada.
<i>Estruturas alternativas de acesso à informação</i>	√	Os títulos das páginas permitem que um navegador para hiperdocumentos HMBS gere um índice a partir do qual o usuário inicia a navegação.
<i>Ligações temporais</i>	×	Não há notação temporal no modelo.
<i>Ligações dinâmicas</i>	×	Não são suportadas diretamente, mas os títulos das páginas permitem que um sistema de apoio ofereça esse tipo de recursos.
<i>Ligações dentro de componentes</i>	×	Statecharts permitem transições com os conjuntos de origem e destino coincidentes, mas o modelo não possui semântica apropriada para tratar esse caso de forma diferente de uma ação de reiniciar a visita.

Requisito	Satisfeito pelo HMBS	Observação
<i>Contextos de informação</i>	√	O uso de hierarquias no statechart permite a especificação desse requisito.
<i>Visão da hierarquia do hiperdocumento</i>	√	A definição do nível de visibilidade e de Show-hview permite a especificação desse requisito.

Tabela 3.1 - Modelo HMBS frente aos requisitos de aplicações hipermédia

3.5 Considerações finais

Este capítulo apresenta o estado da arte na área de especificação de aplicações de multimídia e hipermédia. Isso garante o embasamento teórico necessário à definição de um novo modelo, do qual se espera que, principalmente, as características relativas aos requisitos de sincronização sejam satisfeitas. Além disso, o modelo HMBS, ponto de partida para a criação desse novo modelo, é descrito e discutido detalhadamente em termos dos requisitos de aplicações hipermédia considerados no trabalho, de forma que os seus pontos fracos foram identificados para serem atacados por novas soluções.

Os próximos capítulos apresentam a extensão do HMBS, cujas novas características foram incorporadas visando a eliminar ou, ao menos, minimizar as deficiências identificadas na seção 3.4.4.

Capítulo 4

Hypercharts

4.1 Especificação em alto nível baseada em statecharts

Os *hypercharts* são uma proposta de linguagem de alto nível baseada em statecharts para a especificação de hiperdocumentos. Dessa forma, o principal objetivo dessa linguagem é oferecer um conjunto de notações que permita especificar concisamente, e com as mesmas noções de statecharts, o comportamento de hiperdocumentos complexos, principalmente aqueles que envolvam requisitos de sincronização de dados multimídia em ambientes distribuídos.

A idéia por trás dos *hypercharts* é a de que alguns novos elementos de notação – principalmente aqueles voltados à especificação temporal – podem ter sua semântica descrita por um statechart convencional. Assim, o desenvolvedor tem disponíveis blocos de construção intrínsecos à linguagem, os quais podem ser utilizados como componentes notacionais, sem que o desenvolvedor tome conhecimento deles. Isso causa a diminuição do esforço de desenvolvimento de aplicações.

No contexto do modelo XHMBS, os *hypercharts* substituem os statecharts como técnica de especificação subjacente. Dessa forma, as extensões ao modelo HMBS podem ser classificadas em dois níveis: um relativo ao statechart, que deixa de ser utilizado para dar lugar aos *hypercharts*, com maior poder de especificação temporal e de sincronização, e outro relativo ao próprio modelo que, como é apresentado no capítulo 5, recebeu extensões visando a flexibilizar a integração entre estrutura e conteúdo da informação, assim como a especificação da apresentação do hiperdocumento.

Os *hypercharts* podem ser considerados um superconjunto dos statecharts, de forma que, além de possuírem todas as notações permitidas em statecharts convencionais, introduzem novas notações que permitem a especificação do comportamento temporal de uma aplicação hipermídia. Em termos gerais, as extensões contidas em *hypercharts* são: relógios locais e registradores de história temporal; transições temporais; mecanismos de sincronização; parametrização de estados e abstração de transições. Cada uma dessas extensões é apresentada em mais detalhes nas seções seguintes.

4.2 A base temporal dos *Hypercharts*

Os *hypercharts* possuem mecanismos específicos para a manipulação do tempo na especificação. Uma vez que *hypercharts* constituem uma linguagem de mais alto nível baseada em statecharts, todas as expressões criadas em nível de notação são transformadas em statecharts convencionais equivalentes em um nível semântico. Obviamente, o especificador, utilizando *hypercharts*, não precisa se preocupar com essas transformações, uma vez que as mesmas são intrínsecas à linguagem.

A opção de se utilizar statecharts convencionais para a definição da semântica dos *hypercharts* se deve ao fato de que statecharts temporais [Pnu92], que também permitem especificação temporal, são restritos no que diz respeito à utilização de transições interníveis, referências a nomes de estados e mecanismos de história, além de permitirem o uso de estados instantâneos, ou seja, estados que são ativados e desativados dentro de um mesmo passo de execução [Bee94]. A semântica dos *hypercharts* é baseada na semântica operacional de statecharts apresentada em [Har87a], caracterizada pelos passos. Nessa semântica, um passo é composto de vários micro-passos, dentro dos quais somente uma transição é avaliada, permitindo, assim, a evolução das várias máquinas de estado concorrentemente ativas. Essa semântica permite que um componente OR do statechart evolua somente uma vez a cada passo.

Normalmente, alguma unidade de tempo é determinada como sendo diretamente proporcional a um passo, o que possibilita a representação da passagem do tempo em uma execução ou simulação do statechart. Assim, assume-se a existência de um relógio global para o objeto em especificação (a aplicação hiperfórmula), o qual irá representar a evolução do tempo em incrementos de uma unidade equivalente a um passo.

Adicionalmente, nesta extensão, são definidos, para cada estado do statechart, dois mapeamentos para contadores, representando, respectivamente, um *relógio local* ao estado e um *registrador de história temporal* desse estado. O *relógio local* registra o número de passos durante os quais o estado permanece ativo, sendo incrementado a cada passo. O valor do *relógio local* de um estado é útil para operações de troca de informações temporais entre estados e *expressões de verificação temporal*. O *registrador de história temporal* mantém o último tempo de ativação de um estado, complementando o mecanismo de história nos *hypercharts*. Os mapeamentos e as operações definidas sobre eles são definidos formalmente como segue:

Seja S o conjunto de estados do statechart e N o conjunto dos números naturais. São definidos os mapeamentos:

LSC: $S \rightarrow N$ (Local Step Counter - equivalente, em termos de passos de execução, ao relógio local de um estado)

T_hist: $S \rightarrow N$ (registrador de história temporal de um estado)

e as seguintes operações:

Sejam $s \in S$ e $n \in N$

New_SC_value (s, n) =

$$\left\{ \begin{array}{l} \text{Se, inicialmente, } LSC(s) = v \mid v \in N \\ \text{então } LSC' = (LSC \setminus \{s \rightarrow v\}) \cup \{s \rightarrow n\} \end{array} \right\}$$

New_SC_value (s, n) e LSC (s) := n são notações equivalentes para a mesma operação.

New_Thist_value (s, n) =

$$\left\{ \begin{array}{l} \text{Se, inicialmente, } T_hist(s) = v \mid v \in N \\ \text{então } T_hist' = (T_hist \setminus \{s \rightarrow v\}) \cup \{s \rightarrow n\} \end{array} \right\}$$

New_Thist_value (s, n) e $T_hist(s) := n$ são notações equivalentes para a mesma operação.

Com a definição dessas operações sobre os mapeamentos é possível definir os mecanismos de funcionamento dos relógios locais e de história temporal, descritos a seguir.

4.2.1 História temporal

A notação de *história temporal* permite a associação de um símbolo especial de história às transições. Esse símbolo, um pequeno relógio, associado a uma transição t que tem como destino um estado s indica que, se s já esteve ativo anteriormente, além da recuperação da última configuração de s (em termos de seus subestados), será recuperado também o valor do *registrador de história temporal* desse estado. Analogamente aos símbolos de história definidos em statecharts convencionais [Har87a, Har87b], os símbolos de *história temporal* podem agir recursivamente, recuperando os registradores de história temporal de todos os estados que forem ativados nos níveis abaixo de s . A figura 4.1 mostra um exemplo da notação gráfica de *história temporal*, tanto em um nível (4.1-a), quanto em profundidade (4.1-b).

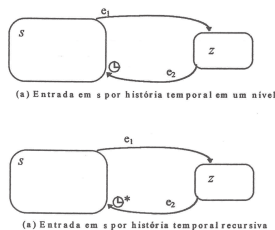


Figura 4.1 - Notação de história temporal

O mecanismo que suporta essa notação é definido da seguinte maneira:

Seja T o conjunto de transições de um *hyperchart*,

$\tau: T \rightarrow \{true, false\} \mid \tau(t) = true$ indica que a transição tem *história temporal* associada e $\tau(t) = false$ indica que a transição não tem *história temporal* associada.

$\tau^*: T \rightarrow \{true, false\} \mid \tau^*(t) = true$ indica que a transição tem *história temporal recursiva* associada e $\tau^*(t) = false$ indica que a transição não tem *história temporal recursiva* associada.

Adicionalmente, sempre que $\tau(t) = true$ admite-se que t entra no estado de destino por história convencional. Analogamente, sempre que $\tau^*(t) = true$, admite-se que t entra no estado de destino por história recursiva convencional.

A todo estado $s \in S \mid \exists t \in T, t = (X, l, Y) \wedge (s \in X) \wedge (\tau(t) = true \vee \tau^*(t) = true)$, está associada a ação "On exit $T_hist(s) := LSC(s)$ ", garantindo que o *registrador de história temporal* será atualizado sempre que o sistema sair de s . Se existe uma transição tal que $\tau(t) = true$, então, quando t disparar, desde que os estados pertencentes ao destino da transição já tenham estado ativos uma vez, a operação $LSC(s_i) := T_hist(s_i)$ é executada para todo estado s_i ativado por história convencional. Para o caso em que $\tau^*(t) = true$, a operação $LSC(s_i) := T_hist(s_i)$ é executada para cada estado s_i ativado por história convencional recursiva. A semântica da recuperação de configurações por história convencional está disponível nos statecharts convencionais.

No caso em que $\tau(t) = false$ e $\tau^*(t) = false$, a operação $LSC(s_i) := 0$ é executada para todo estado s_i recuperado por história convencional ou ativado por "default", ou seja, se não houver recuperação da configuração anterior por *história temporal*, mesmo que haja por história convencional, os relógios locais dos estados recém-ativados são zerados.

Finalmente, o mecanismo que permite fazer com que o *relógio local* de um estado s seja incrementado e, assim, retrate o tempo de ativação de s em um determinado instante é baseado no relógio global do sistema. Assim, quando o sistema tem sua execução iniciada, a operação $LSC(s) := 0$ é executada para todo estado $s \in S$. Adicionalmente, a cada passo p , a operação $LSC(s) := LSC(s) + 1$ é executada para todos os estados que estiverem ativos no passo $p-1$.

Esses mecanismos garantem a base de uma notação mais poderosa e flexível para a observação e manipulação do comportamento temporal da especificação. A partir desses mecanismos e de outros recursos notacionais apresentados adiante, é possível criar especificações complexas de forma bastante simples, sem perda do poder de especificação dos statecharts, uma vez que toda notação disponível em *hypercharts* será convertida, em nível semântico, para statecharts convencionais. A seguir é apresentada a notação para *verificação temporal*.

4.2.2 Verificação temporal

A partir dos mecanismos temporais disponibilizados pelos *hypercharts* para manipulação de tempo (*relógios locais*, *registradores de história temporal* e respectivas operações de atualização) criou-se uma operação lógica que permite a verificação do número de passos (que representa o tempo) de ativação de um determinado estado de forma análoga à verificação de ativação de estados provida pela condição especial *in (state)*.

A operação é denominada *Time-In* (abreviadamente, t_in) e é definida a seguir:

$t_in: S \times N \rightarrow \{true, false\} \mid$

$t_in(state, time) = true \Leftrightarrow LSC(state) \geq time \wedge in(state) = true.$

$t_in(state, time) = false \Leftrightarrow LSC(state) < time \vee in(state) = false.$

A expressão t_in deve ser utilizada de forma análoga a in , como condição de guarda em transições. Esse recurso complementa o uso de “time-outs” e *transições temporais* (apresentadas em seguida), uma vez que a validade de seu valor de retorno pode ser verificada continuamente, o que não ocorre com os “time-outs” que são definidos como eventos e são, portanto, discretos. Dessa forma, utilizando-se t_in torna-se possível verificar se o sistema já esteve determinado tempo em um estado, mesmo que esse estado esteja ativo há mais tempo. Alguns requisitos relacionados com hipermédia ativa, como condições sobre ligações temporais, podem ser facilmente modelados com o uso desse recurso.

É importante notar que, devido à possibilidade de utilização dos mecanismos de história temporal, a semântica da operação de *verificação temporal* possui um detalhe bastante sutil: a possibilidade de que a reativação do estado por história temporal faça com que a condição $t_in(s, t)$ retorne *true* mesmo se o estado tiver sido desativado anteriormente, quando o valor de seu relógio local era menor do que t . Essa implicação é devida à própria semântica desejada para o mecanismo de história temporal. Dessa forma, pode-se dizer que nem sempre é necessário que um estado permaneça continuamente ativo por ao menos t passos para que uma expressão $t_in(s, t)$ retorne valor verdadeiro.

4.3 Transições temporais

Os *hypercharts*, assim como statecharts temporais [Pnu92], possuem recursos de notação temporal que permitem a especificação de transições cujos disparos são determinados pela passagem do tempo enquanto seus estados de origem estão ativos. Com a notação temporal dos *hypercharts* é possível especificar requisitos de sincronização e comunicação como atraso e “jitter”, além de viabilizar o mecanismo que modela a apresentação da informação no modelo XHMBBS.

Os *eventos temporais* têm a forma “[α, η, β]” e não podem ser relacionados com outros eventos ou condições em uma mesma transição. Os rótulos das transições, no entanto, continuam podendo possuir expressões de ações. Além da restrição quanto à combinação de eventos e condições em um mesmo rótulo, as *transições temporais* devem possuir cardinalidade “um para um”, ou seja, devem envolver somente um estado de origem e um de destino. Além disso, não é permitido que um mesmo estado seja origem de mais do que uma transição temporal, uma vez que o uso de várias transições temporais partindo de um mesmo estado de origem equivale à combinação lógica “ou” dos *eventos temporais* contidos nos rótulos dessas transições, o que não é permitido pela sintaxe da linguagem. Essa restrição sintática não restringe a capacidade de especificação da técnica, visto que, como os *eventos temporais* especificam intervalos de tempo, sempre é possível obter um intervalo equivalente a uma união ou intersecção de vários intervalos.

Formalmente, se E é o conjunto de eventos de um *hyperchart*, incluindo todas as definições de eventos dos statecharts, então

$$(\alpha, \eta, \beta \in Q^* \mid \alpha \leq \eta \leq \beta) \Leftrightarrow [\alpha, \eta, \beta] \in E$$

Seja S o conjunto de estados e L o conjunto de rótulos de transições de um *hyperchart*.

Seja $T \subset 2^S \times L \times 2^{S \times H}$ o conjunto das transições de um *hyperchart*.

Dado $t = (A, l, B) \in T$

t é uma transição temporal se l é da forma “[α, η, β] / a ”, com $\#(A) = 1, \#(B) = 1$ e se $\neg (\exists t' \in T \mid t' = (A, "[\alpha', \eta', \beta'] / a', \{s\}))$.

A figura 4.2 apresenta a forma genérica de uma transição com *evento temporal* associado. O uso de uma *transição temporal* nesse exemplo implica no fato de que há uma *atividade principal* A sendo controlada pelo estado X (*atividade de apresentação de X*), a qual possui um evento end_X associado, relativo ao seu término. A atividade A pode ser nula, o que implica no fato de que o evento end_X irá ocorrer assim que o estado se tornar ativo. *Atividades nulas* são utilizadas com estados que não representam, efetivamente, o comportamento de qualquer funcionalidade na especificação. Estados desse tipo são utilizados, normalmente, para denotar apenas a passagem de tempo.

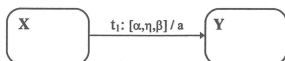


Figura 4.2 - Transição temporal

A notação $[\alpha, \eta, \beta]$ impõe duas restrições ao disparo da transição t_1 . A primeira é que, mesmo que a atividade A tenha terminado, isto é, que end_X tenha ocorrido, se o *limite temporal mínimo* α não tiver sido atingido, ou seja, se o sistema não permaneceu ao menos α passos no estado X (respeitadas as implicações semânticas relativas ao uso de *história temporal*), t_1 não disparará. Nesse caso, t_1 disparará assim que o limite mínimo for atingido. Caso o evento de término end_X ocorra em algum momento relativo entre α e β , t_1 irá disparar imediatamente. A segunda restrição é relativa ao *limite temporal máximo* β . Se o sistema permanecer no estado X até o momento relativo β e o evento end_X não ocorrer, ou seja, se a atividade A não terminar a tempo, t_1 irá disparar de qualquer maneira, abortando a execução de todas as atividades associadas ao estado X , inclusive, e principalmente, A .

Como visto, a semântica pretendida para as transições temporais está associada com a passagem do tempo (modela pelos passos de execução) e com a ocorrência de um evento que indica o final de uma atividade associada ao estado internamente. No caso do uso de *hypercharts* com o XHMBS, essa atividade é a apresentação da página associada ao estado pelo canal de apresentação correspondente. A passagem do tempo é modelada semanticamente através de uma combinação de subestados intermediários com eventos de “time-out”. O evento indicador de término da atividade é um evento primitivo definido para o estado de origem da transição temporal.

Para garantir o funcionamento combinado dos mecanismos de *história temporal* e de *transições temporais* dos *hypercharts*, o evento de “time-out” recebeu uma notação adicional

para tratar da permanência em um estado. Usualmente o evento de "time-out" é representado por "tm (*ev*, *n*)", onde *ev* é um evento e *n* é o número de passos decorridos após a ocorrência de *ev*. Assim, quando se quer especificar um evento de "time-out" associado a um determinado tempo de permanência em um estado *s* utiliza-se a expressão "tm (*en(s)*, *n*)", onde *en(s)* é o evento que ocorre quando da entrada no estado *s*. O uso desse tipo de expressão, porém, restringe a combinação de *história temporal* com *transições temporais*, visto que o mecanismo de *história temporal* é baseado nos relógios locais dos estados, e não na ocorrência de eventos. Dessa forma, os *hypercharts* possuem a notação especial de "tm (*s*, *n*)", onde *s* é um estado e *n* é o número de passos durante os quais o estado *s* permaneceu ativo. A semântica para essa notação é dada por "tr (t_in(*s*, *n*))", ou seja, tm (*s*, *n*) denota o evento que ocorre quando o relógio local do estado *s* atinge o valor *n*. Essa expressão de "time-out" é utilizada para a definição da semântica das *transições temporais* dos *hypercharts*, procurando garantir a sua correitude quando o estado de origem de uma *transição temporal* também for destino de uma *transição* com *história temporal* associada.

A transformação de um diagrama notacional como o da figura 4.2 em um statechart equivalente à sua semântica obedece a três estratégias: uma voltada para estados do tipo AND; uma voltada para estados do tipo OR; e, finalmente, outra voltada para estados básicos. Em todas essas estratégias de transformação, o objetivo é incorporar a *X* um *TCS* ("Time Controller Statechart"), capaz de controlar o comportamento temporal de uma *transição t* com origem em *X*. O *TCS* será sempre um estado concorrente aos subestados de *X* quando $\psi(X) = \text{OR}$ ou $\psi(X) = \text{AND}$. Se $\rho(X) = \emptyset$, então o *TCS* será o próprio estado *X*. As estratégias de transformação semântica são apresentadas a seguir.

Estados do tipo AND

Considerando o exemplo da figura 4.3-a, em que $\psi(X) = \text{AND}$, então será criado um estado X_{f1_TCS} tal que $X_{f1_TCS} \in \rho(X)$; X_{f1_TCS} é o *TCS* associado ao estado *X* e à *transição t*. Assim, se, inicialmente, $\rho(X) = \{X_1, X_2, \dots, X_n\}$, tem-se que, após a transformação semântica, $\rho(X) = \{X_1, X_2, \dots, X_n, X_{f1_TCS}\}$.

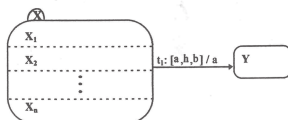


Figura 4.3(a) - Exemplo de transição temporal associada a um estado do tipo AND

Depois de transformado o *hyperchart* da figura 4.3-a será um statechart como o que é apresentado na figura 4.3-b. Deve-se notar que, além dos subestados do *TCS* incorporado, os quais serão sempre estruturados da mesma forma, assim como as respectivas *transições*, foram criadas as variáveis lógicas *end_X_f1*, que indica a ocorrência do evento *end_X* (término da atividade); *max_X_f1*, que indica que a *transição temporal* disparou devido ao fato de o *limite temporal máximo* ter sido atingido e o evento de fim não ter ocorrido no intervalo $[\alpha, \beta]$; e *min_X_f1*, que indica que a permanência no estado *X* atingiu o *limite temporal mínimo*.

especificado pela transição temporal. Essas variáveis são utilizadas na semântica dos *hypercharts* para a definição dos *mecanismos de sincronização*.

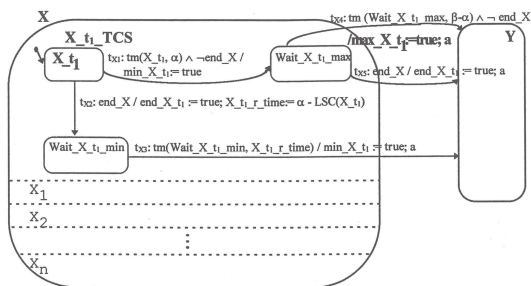


Figura 4.3(b) - Exemplo de transformação para estados do tipo AND

É importante observar que os nomes das variáveis e eventos são criados de acordo com os identificadores do estado e da transição. Para efeito de clareza do exemplo, neste caso usa-se o nome do estado (X) e um identificador extraído do rótulo da transição (" t_i "). Assim, por exemplo, $max_X_t_i$ refere-se à indicação de saída do estado X devido ao fato de que o *limite temporal máximo* da transição t_i foi atingido. No entanto, como se trata de um processo de transformação da notação em nível sintático para um equivalente semântico, pode-se assumir que esses identificadores são criados internamente, ficando o desenvolvedor isento da responsabilidade de prover os identificadores para esse fim em sua especificação. Os nomes de estado continuam sendo uma necessidade sintática imposta pela notação de statecharts.

A forma genérica da geração do TCS para um estado X do tipo AND do qual parte uma *transição temporal* t pode ser formalizada da seguinte forma:

Sejam S o conjunto de estados de um *hyperchart*, T o conjunto de suas transições, L o conjunto de rótulos das suas transições, V o conjunto de suas expressões e C o conjunto de suas condições.

se $\rho(X) \neq \emptyset \wedge \psi(X) = \text{AND}$, então

dado $t = (\{X\}, l, \{Y\}) \mid l = "[\alpha, \eta, \beta] / a"$, $t \in T$

Tem-se que

$S' = S \cup \{X_t_TCS\} \mid X_t_TCS \in \rho(X) \wedge \psi(X_t_TCS) = \text{OR} \wedge \psi(X) = \text{AND}$.

$S' = (S \cup \text{Sub_TCS}) \mid \text{Sub_TCS} = \{X_t, \text{Wait_X_t_min}, \text{Wait_X_t_max}\} \wedge \rho(X_t_TCS) = \text{Sub_TCS}$

$$V' = V \cup \{\text{min_X_t}, \text{end_X_t}, \text{max_X_t}, \text{X_t_r_time}\}$$

$$E' = E \cup \{\text{end_X}\}$$

$$L' = L \cup \{l_1, l_2, l_3, l_4, l_5\}$$

$$l_1 = \text{"tm}(X_t, \alpha) \wedge \neg \text{end_X} / \text{min_X_t} := \text{true}",$$

$$l_2 = \text{"end_X} / \text{end_X_t} := \text{true}; \text{X_t_r_time} := \alpha - \text{LSC}(X_t)",$$

$$l_3 = \text{"tm}(\text{Wait_X_t_min}, \text{X_t_r_time}) / \text{min_X_t} := \text{true}; a",$$

$$l_4 = \text{"tm}(\text{Wait_X_t_max}, \beta - \alpha) \wedge \neg \text{end_X} / \text{max_X_t} := \text{true}; a",$$

$$l_5 = \text{"end_X} / \text{end_X_t} := \text{true}; a"$$

$$T' = T \cup \{tx_1, tx_2, tx_3, tx_4, tx_5\}$$

$$tx_1 = (\{X_t\}, l_1, \{\text{Wait_X_t_max}\}),$$

$$tx_2 = (\{X_t\}, l_2, \{\text{Wait_X_t_min}\}),$$

$$tx_3 = (\{\text{Wait_X_t_min}\}, l_3, \{Y\}),$$

$$tx_4 = (\{\text{Wait_X_t_max}\}, l_4, \{Y\}),$$

$$tx_5 = (\{\text{Wait_X_t_max}\}, l_5, \{Y\})$$

O estado X_t_TCS recebeu a atividade "On entry end_X_t := false; min_X_t := false; max_X_t := false".

A estruturação de um TCS, com ⁵⁷ dito anteriormente, visa a modelagem do comportamento de uma *transição temporal* t que tem origem em um estado X qualquer. Cada uma das cinco transições definidas em um TCS (tx_i , $1 \leq i \leq 5$) modela a ocorrência de um evento relativo ao comportamento temporal de t . Assim, a transição tx_1 modela o alcance do *limite temporal mínimo* quando a aplicação estiver no subestado X_t , o qual modela o estado inicial de uma *transição temporal*. Caso o *limite temporal mínimo* de t não seja alcançado e o evento de término da *atividade de apresentação* de X (end_X) ocorra, então a transição tx_2 leva a aplicação ao estado Wait_X_t_min . A aplicação permanecerá nesse estado pelo tempo (ou, mais precisamente, pelo número de passos) complementar ao *limite temporal mínimo*, calculado na expressão de ações de tx_2 . Dessa forma, quando o *limite temporal mínimo* de t for atingido, a transição tx_3 disparará e levará a aplicação ao conjunto de estados de destino especificado por t . Se, no entanto, o evento de término da *atividade de apresentação* de X ocorrer depois que a aplicação estiver no estado Wait_X_t_max , as transições tx_4 e tx_5 , modelam o disparo de t , respectivamente, pelo alcance do *limite temporal máximo*, ou pela ocorrência do evento de término da *atividade de apresentação* de X entre os *limites temporais* da transição, levando a aplicação, de qualquer forma, ao conjunto de estados de destino de t . Além dos estados e transições, as variáveis criadas em um TCS indicam, para verificações posteriores, como se deu o comportamento de t .

Estados do tipo OR

Considerando o exemplo da figura 4.4-a, em que $\psi(X) = \text{OR}$, então serão criados os estados X e X_t_TCS , tais que $\rho(X) = \rho(X)$ e X_t_TCS é o TCS associado ao estado X e à transição t_1 . Além disso, $\psi(X) = \text{OR}$ e $\rho(X) = \{X, X_t_TCS\}$. Assim, se, inicialmente, $\rho(X) =$

$\{X_1, X_2, \dots, X_n\}$, tem-se que, após a transformação semântica, $\rho(X) = \{X', X_{t_TCS}\}$, $\rho(X') = \{X_1, X_2, \dots, X_n\}$ e $\psi(X) = \text{AND}$.

A figura 4.4-a apresenta um *hyperchart* baseado no exemplo da figura 4.2. A figura 4.4-b apresenta a mesma especificação transformada em um statechart equivalente.

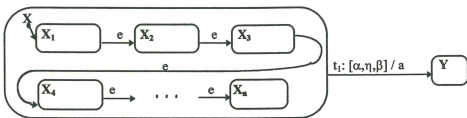


Figura 4.4(a) - Exemplo de transição temporal associada a um estado do tipo OR

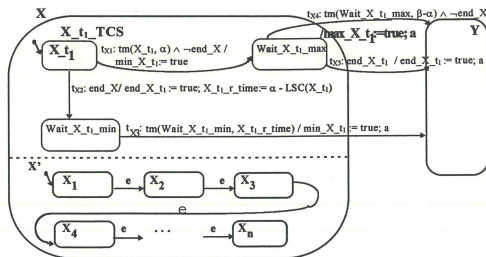


Figura 4.4(b) - Exemplo de transformação para estados do tipo OR

Dessa forma, a geração do TCS para um estado X do tipo OR do qual parte uma transição temporal t é formalizada como segue:

Sejam S o conjunto de estados de um *hyperchart*, T o conjunto de suas transições, L o conjunto de rótulos das suas transições, V o conjunto de suas expressões e C o conjunto de suas condições.

se $\rho(X) \neq \emptyset \wedge \psi(X) = \text{OR}$, então
 dado $t = (\{X\}, l, \{Y\}) \mid l = "[\alpha, \eta, \beta] / a", t \in T$

Tem-se que

Seja $\text{Inic} = \rho(X)$

$S' = S \cup \{X', X_{t_TCS}\} \mid \rho(X') = \text{Inic} \wedge \rho(X) = \{X', X_{t_TCS}\} \wedge \psi(X') = \text{OR} \wedge \psi(X) = \text{AND}$.



$S^* = (S \cup \text{Sub_TCS}) \mid \text{Sub_TCS} = \{X_t, \text{Wait_X_t_min}, \text{Wait_X_t_max}\} \wedge \rho(X_t_TCS)$
 $= \text{Sub_TCS}$

$V^* = V \cup \{\text{min_X_t}, \text{end_X_t}, \text{max_X_t}, \text{X_t_r_time}\}$

$E^* = E \cup \{\text{end_X}\}$

$L^* = L \cup \{l_1, l_2, l_3, l_4, l_5\}$

$l_1 = \text{"tm}(X_t, \alpha) \wedge \neg \text{end_X} / \text{min_X_t} := \text{true}"$,

$l_2 = \text{"end_X} / \text{end_X_t} := \text{true}; \text{X_t_r_time} := \alpha - \text{LSC}(X_t)"$,

$l_3 = \text{"tm}(\text{Wait_X_t_min}, \text{X_t_r_time}) / \text{min_X_t}_1 := \text{true}; a"$,

$l_4 = \text{"tm}(\text{Wait_X_t_max}, \beta - \alpha) \wedge \neg \text{end_X} / \text{max_X_t} := \text{true}; a"$,

$l_5 = \text{"end_X} / \text{end_X_t} := \text{true}; a"$

$T^* = T \cup \{tx_1, tx_2, tx_3, tx_4, tx_5\}$

$tx_1 = (\{X_t\}, l_1, \{\text{Wait_X_t_max}\})$,

$tx_2 = (\{X_t\}, l_2, \{\text{Wait_X_t_min}\})$,

$tx_3 = (\{\text{Wait_X_t_min}\}, l_3, \{Y\})$,

$tx_4 = (\{\text{Wait_X_t_max}\}, l_4, \{Y\})$,

$tx_5 = (\{\text{Wait_X_t_max}\}, l_5, \{Y\})$

O estado X_t TCS recebeu a atividade "On entry end_X_t := false; min_X_t := false; max_X_t := false".

Estados básicos

Se o estado X da figura 4.2 for um estado básico, ou seja, se $\rho(X) = \emptyset$, o TCS correspondente à transição t_1 será encapsulado por X . Dessa forma, ocorrerá que o próprio estado X será um TCS. Essa transformação pode ser observada na figura 4.5.

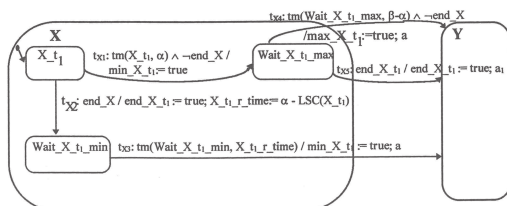


Figura 4.5 - Exemplo de transformação para estados básicos

A definição formal do caso genérico para a geração do TCS para um estado básico X do qual parte uma *transição temporal* t é dada a seguir:

Sejam S o conjunto de estados de um *hyperchart*, T o conjunto de suas transições, L o conjunto de rótulos das suas transições, V o conjunto de suas expressões e C o conjunto de suas condições.

se $\rho(X) = \emptyset$, então

Dado $t = (\{X\}, l, \{Y\}) \mid l = "[\alpha, \eta, \beta] / a"$ $\mid t \in T$, então

$S' = (S \cup \text{Sub_TCS}) \mid \text{Sub_TCS} = \{X_t, \text{Wait_X_t_min}, \text{Wait_X_t_max}\} \wedge \rho(X) = \text{Sub_TCS} \wedge \psi(X) = \text{OR}$

$V = V \cup \{\text{min_X_t}, \text{end_X_t_ok}, \text{max_X_t}, \text{X_t_r_time}\}$

$E' = E \cup \{\text{end_X}\}$

$L' = L \cup \{l_1, l_2, l_3, l_4, l_5\} \mid$

$l_1 = \text{"tm}(X_t, \alpha) \wedge \neg \text{end_X} / \text{min_X_t} := \text{true}"$,

$l_2 = \text{"end_X} / \text{end_X_t} := \text{true}; \text{X_t_r_time} := \alpha - \text{LSC}(X_t)"$,

$l_3 = \text{"tm}(\text{Wait_X_t_min}, \text{X_t_r_time}) / a"$,

$l_4 = \text{"tm}(\text{Wait_X_t_max}, \beta - \alpha) \wedge \neg \text{end_X} / \text{max_X_t} := \text{true}; a"$,

$l_5 = \text{"end_X} / \text{end_X_t} := \text{true}; a"$

$T' = T \cup \{tx_1, tx_2, tx_3, tx_4, tx_5\} \mid$

$tx_1 = (\{X_t\}, l_1, \{\text{Wait_X_t_max}\})$,

$tx_2 = (\{X_t\}, l_2, \{\text{Wait_X_t_min}\})$,

$tx_3 = (\{\text{Wait_X_t_min}\}, l_3, \{Y\})$,

$tx_4 = (\{\text{Wait_X_t_max}\}, l_4, \{Y\})$,

$tx_5 = (\{\text{Wait_X_t_max}\}, l_5, \{Y\})$

O estado X recebeu a atividade "On entry end_X_t := false; min_X_t := false; max_X_t := false".

Com as *transições temporais*, os *hypercharts* ganham poder de especificação temporal, dotando a linguagem de um recurso importante para a especificação de comportamentos que se caracterizam por progredirem independentemente de interações com o ambiente da aplicação. É possível, no entanto, que transições rotuladas por eventos convencionais, ou até mesmo de "time-out", sejam utilizadas em conjunto com *transições temporais*, disponibilizando um mecanismo de especificação de *preempção* bastante simples.

4.4 Mecanismos de sincronização

Quando, recursivamente, especifica-se que vários *hypercharts* devem estar ativos concorrentemente é desejável que existam mecanismos de sincronização, de forma que esses *hypercharts* possam se comunicar. Um desses mecanismos é suportado pela difusão de

eventos, ou "broadcasting". Nesse caso, a ação de uma transição pode corresponder à geração um evento interno, o qual será sentido por todos os *hypercharts* ativos concorrentemente. Essa noção é exatamente a mesma existente em statecharts.

A geração de eventos internos é um recurso de sincronização bastante poderoso, mas, principalmente para a modelagem de aplicações hipermídia, em que a noção de duração é algo fundamental, nota-se que esse mecanismo não é suficiente, uma vez que podem existir situações em que seja necessário que um sinal de sincronização perdure por algum tempo, até que os *hypercharts* concorrentes possam ser re-sincronizados. Ainda assim, alguns casos particulares de sincronização de dados multimídia podem ser especificados com o uso de ações combinadas com *transições temporais* em statecharts. Essa solução já mostrou ter melhorado o poder de especificação do modelo HMBS, mas ainda é limitada [Pau96].

Outro recurso de sincronização disponível em statecharts e, portanto, herdado pelos *hypercharts* é a notação de transições M:N, as quais envolvem vários estados na origem e no destino. Novamente, para o caso de aplicações hipermídia, muitas vezes o impedimento de utilizar vários rótulos, cada qual associado a um arco que parte de um estado de origem, pode significar uma restrição, visto que, na especificação dessas aplicações, além de se considerar a ativação dos estados de origem, pode-se querer considerar as restrições temporais associadas a eles, as quais podem ser dadas por *eventos temporais*.

Em face dessas limitações, os *hypercharts* disponibilizam a notação de *transições M:N sincronizadas*, inspirada nos modelos TSPN [Sen94] e TSPN_{UI} [Coo95], baseados em redes de Petri. O mecanismo em questão é baseado na idéia de transições M:N, mas possui notação própria, permitindo que os arcos que partem dos estados de origem sejam rotulados com *eventos temporais* e que um *tipo* (ou regra) de sincronização seja associado à transição.

Uma *transição M:N sincronizada* possui cinco grupos de componentes: *estados de origem*, *arcos de origem*, *estados de destino*, *rótulos* (associados a cada *arco de origem*) e um *tipo de sincronização*. O *tipo* anotado em uma *transição M:N sincronizada* determina como as restrições impostas pelos *eventos temporais* associadas a cada *arco de origem* devem ser tratadas em termos da geração de uma restrição temporal única para toda a transição. Dessa forma, diz-se que os *tipos* de sincronização definem a semântica de sincronização entre os *hypercharts* concorrentes envolvidos ou, em termos de aplicações hipermídia, da sincronização entre cadeias de dados. A forma geral da *transição M:N sincronizada* é apresentada na figura 4.6.

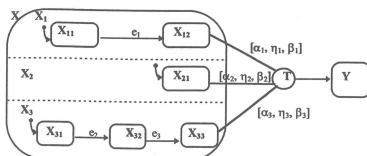


Figura 4.6 - Notação genérica da *transição M:N sincronizada* (com $N=1$)

Os tipos de sincronização utilizados nas transições $M:N$ sincronizadas são os mesmos utilizados pelos modelos TSPN e $TSPN_{0n}$, mantendo o mesmo significado que têm nesses modelos, de forma que o processo de transformação semântica leva em consideração as regras de disparo de transições tipadas estabelecidas nesses modelos. A semântica dessas regras de disparo leva em consideração as cadeias de dados envolvidas com a transição tipada, gerando um *intervalo dinâmico de disparo* para essa transição. Em termos de *hypercharts*, são considerados os componentes concorrentes aos quais pertencem os *estados de origem* da transição. Para manter a consistência com a sintaxe das *transições temporais* e com regras de disparo propostas pelos modelos TSPN e $TSPN_{0n}$, é imposta a restrição os *arcos de origem*, não devem possuir condições ou ações em seus rótulos. Em outras palavras, a única expressão permitida em um rótulo de um *arco de origem* de uma transição $M:N$ sincronizada é um único *evento temporal*.

Dado uma transição $M:N$ sincronizada, podem existir nove tipos de sincronização a serem aplicados sobre os componentes envolvidos: *strong-or*, *weak-and* e *master*; *or* e *and* os cinco tipos básicos, e *or-master*, *strong-master*, *weak-master* e *and-master*, os tipos derivados de combinações entre os primeiros. O tipo *master* de sincronização considera as restrições temporais associadas a um único *hyperchart* envolvido na sincronização, o qual é definido pelo especificador como o "master" daquela transição. Os tipos, representados genericamente pela letra T ("type") na figura 4.6, são abreviados da seguinte forma:

Strong-or	S O	Weak-and	W A	Master	M (s)
Or	OR	And	A	Or-Master	O M (s)
Strong-Master	S M (s)	Weak-Master	W M (s)	And-Master	A M (s)

Nessa notação, s é um *estado de origem* da transição, pertencente ao *hyperchart* considerado "master". A semântica adotada para as transições $M:N$ sincronizadas é dada, a exemplo das *transições temporais*, pela transformação do *hyperchart* em um statechart convencional equivalente, o qual representa, em termos de seus estados e transições, o *intervalo dinâmico de disparo* da transição conforme o tipo especificado.

Formalmente, o conjunto Ts de transições $M:N$ sincronizadas de um *hyperchart* pode ser definido da seguinte maneira:

$$Ts \subseteq (2^{S_{cl}}) \times 2^{S_{ch}} \times sync \times (S \cup \emptyset), \text{ em que}$$

$$sync = \{SO, WA, M, OR, A, SM, WM, OM, AM\}$$

Uma transição $M:N$ sincronizada $ts = (\text{Source}, \text{Target}, \text{syn}, s)$ com $\text{Source} = \{<s_i, l_i \mid s_i \in S \wedge l_i \in L \wedge 1 \leq i \leq M\}$ está sujeita às seguintes restrições:

$$M \geq 2;$$

$$syn \in \{M, SM, WM, OM, AM\} \Rightarrow s \neq \emptyset$$

Para uma transição $M:N$ sincronizada com n estados de origem, cujos respectivos *arcos de origem* são rotulados com expressões do tipo $[\alpha_i, \eta_i, \beta_i]$ ($1 \leq i \leq n$), tem-se que o

momento relativo θ de seu disparo está contido em um intervalo de tempo determinado pelo tipo da transição, como apresentado a seguir:

$$\textit{Strong-or} - \min_i(\alpha_i) \leq \theta \leq \min_i(\beta_i); \textit{Weak-and} - \max_i(\alpha_i) \leq \theta \leq \max_i(\beta_i);$$

$$\textit{Master}(s_i) - \alpha_i \leq \theta \leq \beta_i; \textit{Or} - \min_i(\alpha_i) \leq \theta \leq \max_i(\beta_i);$$

$$\textit{And} - \max_i(\alpha_i) \leq \theta \leq \max(\min_i(\beta_i), \max_i(\alpha_i)); \textit{Or-Master}(s_i) - \min_i(\alpha_i) \leq \theta \leq \beta_i;$$

$$\textit{Strong-Master}(s_i) - \alpha_i \leq \theta \leq \max(\min_i(\beta_i), \alpha_i);$$

$$\textit{Weak-Master}(s_i) - \alpha_i \leq \theta \leq \max_i(\beta_i);$$

$$\textit{And-Master}(s_i) - \max_i(\alpha_i) \leq \theta \leq \max(\max_i(\alpha_i), \beta_i);$$

Antes de se apresentar as transformações semânticas para cada tipo de transição, é necessário que consideramos que quando uma transição M:N, sincronizada ou não, possui um conjunto S_c de estados de origem, necessariamente se tem que:

$$\exists X \mid S_c \subset \rho^*(X) \wedge \psi(X) = \text{AND}, \text{ ou seja, os estados de origem de uma transição M:N possuem um ancestral comum do tipo AND.}$$

Diante dessa consideração, pode-se dizer, informalmente, que a parte do *hyperchart* a ser transformada quando uma *transição M:N sincronizada* é especificada é o menor ancestral do tipo AND comum a todos os estados de origem daquela transição.

Os tipos de *transições M:N sincronizadas* e as respectivas transformações são apresentados a seguir. Os termos correteidade temporal e cenário de sincronização são frequentemente utilizados na descrição dos tipos de sincronização, merecendo uma melhor definição antes de sua introdução.

Por correteidade temporal de uma cadeia de dados, entende-se a propriedade dessa cadeia que garante que a sua apresentação respeite as restrições temporais impostas sobre ela. Assim, em termos da notação de *hypercharts*, a correteidade temporal de uma página associada a um estado s é garantida se as restrições da transição temporal ou do *arco de origem* que parte de s forem satisfeitas. Se uma cadeia de dados é representada por um conjunto de estados mapeados para páginas e relacionados por transições temporais, a correteidade temporal dessa cadeia é garantida se ela o for para cada estado que compõe essa cadeia.

Por cenário de sincronização entende-se o conjunto de componentes concorrentes do *hyperchart* envolvidos em uma *transição M:N sincronizada*. Dessa forma, um cenário de sincronização sempre envolverá duas ou mais cadeias de dados que serão sincronizadas em algum ponto, cada uma dessas cadeias sendo representada por um componente OR. Normalmente, um cenário de sincronização é determinado por dois pontos de sincronização: um que dá origem ao cenário e outro que determina o seu final.

4.4.1 Sincronização do tipo *Strong-or*

O objetivo do tipo de sincronização *strong-or* é de garantir a corretude temporal da cadeia de dados mais avançada. Em termos da transição em *hypercharts*, trata-se de considerar aquele *estado de origem* que, antes dos demais, tiver sua *atividade de apresentação* encerrada. O encerramento dessa atividade pode ocorrer normalmente ou se o *limite temporal máximo*, especificado pelo *evento temporal* associado ao respectivo *arco de origem*, for alcançado.

Dessa forma, considerando o *hyperchart* da figura 4.7, que denota uma transição *M:N sincronizada* do tipo *strong-or*, tem-se que a representação em nível semântico é da forma apresentada na figura 4.8.

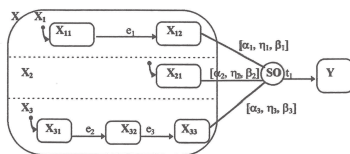
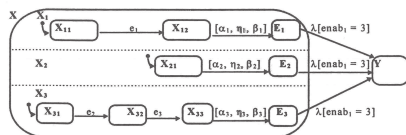


Figura 4.7 - Exemplo de transição *M:N sincronizada* do tipo *strong-or*



O estado X recebeu a atividade:
 Ou entry enab₁ := 0
 Os estados X₁₂, X₂₁ e X₃₃ receberam, cada um, a atividade:
 Ou entry enab₁ := cr(enab₁) + 1

Figura 4.8 - Exemplo de transformação para uma transição do tipo *strong-or*

Na figura 4.8, nota-se a criação da variável *enab₁* ("enabled") e dos estados *E₁*, *E₂* e *E₃*, utilizados para sinalizar o final da execução das atividades de apresentação, respectivamente, dos componentes *X₁*, *X₂* e *X₃*. A variável *enab₁* indica, com seu valor, a quantidade de estados de origem da transição *M:N sincronizada* que estão ativos. Uma atividade associada ao estado X inicia essa variável. Os estados de origem, por sua vez, recebem atividades que incrementam o valor de *enab* quando são ativados.

Nesse tipo de transição, desde que todos os *estados de origem* estejam ativos, aquele *estado de origem* que tiver sua atividade encerrada primeiro causará o disparo. Dessa forma, qualquer das transições definidas no nível semântico (no statechart correspondente) poderá disparar entre o menor dos *limites temporais mínimos* e o menor dos *limites temporais*

máximos, respeitando a semântica original das transições do tipo *strong-or*, dada pelo modelo TSPN.

A condição de disparo das transições ao nível do statechart é dada pela expressão $\lambda[enab_i = 3]$, que checa se os três estados de origem estão ativos. Para o caso genérico, o valor de $enab_i$ correspondente à ativação de todos os estados de origem será igual ao número de componentes concorrentes envolvidos na transição. A condição da transição é o único guarda para o seu disparo, uma vez que λ representa o evento nulo, o qual é gerado em todos os passos.

Em termos teóricos, a necessidade de se verificar a ativação dos *estados de origem* vem da semântica convencional das transições M:N, que exige que todos os seus estados de origem estejam ativos para que ocorra o disparo. Assim, como os statecharts gerados determinam o comportamento de uma transição M:N dada pelo *hyperchart*, a necessidade de verificação de ativação dos *estados de origem* é herdada. Além disso, essa noção de ativação de *estados de origem* garante compatibilidade com a definição das transições nos modelos baseados em Redes de Petri, em que uma transição só é considerada habilitada se todos os lugares de origem possuírem fichas.

O tipo de sincronização *strong-or*, privilegiando as cadeias de dados mais avançadas, não garante a correção da apresentação das cadeias em retardo, fazendo com que essas cadeias tenham suas apresentações interrompidas quando a sincronização for aplicada. Dessa forma, é aconselhável utilizar transições do tipo *strong-or* quando a apresentação total de uma das cadeias de dados for suficiente para expressar o significado de todo o *cenário de sincronização*.

Formalmente, a transformação de uma *transição M:N sincronizada* do tipo *strong-or* pode ser definida como segue:

Dado $t_k = (\text{Source}, \text{Target}, SO, \emptyset)$

$$(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i])", (i \in N \mid 1 \leq i \leq M) \} \wedge \text{Target} \subset 2^{(S \times I)})$$

A transformação de t_k em uma representação intermediária, ou seja, em um *hyperchart* ainda contendo *transições temporais*, é dada por:

$$V' = V \cup \{enab_k\}$$

$$C' = C \cup \{c_k\} \mid c_{so} = "enab_k = M"$$

$$L' = L \cup \{l_k\} \mid l_{so} = "\lambda [c_{so}]"$$

$$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M)$$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i ,

então: $S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$

$$T' = T \cup \{t_{SE_i}, t_{E_i}\} \mid$$

$$t_{SE_i} = (\{s_i\}, l_i, \{E_i\}) \wedge t_{E_i} = (\{E_i\}, l_{so}, \text{Target})$$

O estado s_i recebeu a atividade "**On entry** $enab_k := cr(enab_k) + 1$ "

O estado $AncX = LCA(\{E_i \mid i \in N, 1 \leq i \leq M\})$ recebeu a atividade "On entry $enab_k := 0$ ".

O LCA ("Least Common Ancestor"), ou menor ancestral comum, de um conjunto de estados X é um estado $x \in S$ tal que $X \subseteq \rho^+(x)$ e $s \in \rho^+(x) \Rightarrow X \not\subseteq \rho^+(s)$, ou seja, o estado s é um ancestral comum a todos os estados do conjunto X e, além disso, é entre aqueles estados o de nível mais baixo na hierarquia.

4.4.2 Sincronização do tipo Weak-and

O objetivo do tipo de sincronização *weak-and* é o de garantir a correteza temporal da cadeia de dados mais atrasada. Em termos da transição em *hypercharts*, trata-se de considerar o último estado de origem a ter sua atividade de apresentação encerrada.

Se a transição *M:N sincronizada* da figura 4.7 tivesse o tipo WA associado, em vez de SO, denotando, portanto, uma transição do tipo *weak-and*, a transformação semântica geraria uma representação como a apresentada na figura 4.9.

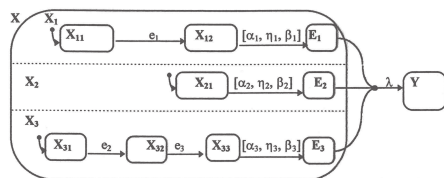


Figura 4.9 - Exemplo de transformação para uma transição do tipo *weak-and*

Na transformação para transições do tipo *weak-and*, como também ocorre com todos os outros tipos de transição, são inseridos somente os estados E_1 , E_2 e E_3 com a mesma finalidade descrita para o tipo *strong-or*. Não há necessidade de uma variável para verificar a ativação dos estados de origem, visto que a condição de disparo é o término das atividades de apresentação de todos os estados. Assim, uma transição *M:N convencional* é utilizada, testando o término de todas as apresentações e, conseqüentemente, a ativação dos estados de origem. Dessa forma, visto que, para essa transformação, o evento nulo é associado a uma transição *M:N convencional*, em nível semântico, o disparo dessa transição ocorre um passo após o último estado de origem da transição *M:N sincronizada* correspondente ter sido desabilitado. Dessa forma, a transição deverá disparar entre o maior limite temporal mínimo e o maior limite temporal máximo das cadeias envolvidas, como definido originalmente para o tipo de *weak-and*.

O tipo *weak-and* é útil para cenários de sincronização em que o significado global é dado pela apresentação completa de todas as cadeias de dados envolvidas. Por outro lado, devido ao fato de privilegiar a cadeia mais atrasada, esse tipo de sincronização faz com que algumas cadeias (as mais avançadas) tenham suas apresentações aparentemente congeladas em suas unidades finais enquanto esperam pelo término das demais.

Em termos formais, essa transformação é expressa como segue:

Dado $t_k = (\text{Source}, \text{Target}, WA, \emptyset)$

$$(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i])", (i \in N \mid 1 \leq i \leq M) \}) \wedge (\text{Target} \subset 2^{S \times H})$$

A transformação de t_k em uma representação intermediária é dada por:

$$L' = L \cup \{l_k\} \mid l_{WA} = "A"$$

$$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i ,

então: $S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$

$$T' = T \cup \{t_{SEi}\} \mid t_{SEi} = (\{s_i\}, l_i, \{E_i\})$$

$$T'' = T \cup \{t_{MNk}\} \mid t_{MNk} = (\{E_i \mid i \in N, 1 \leq i \leq M\}, l_{WA}, \text{Target}).$$

4.4.3 Sincronização do tipo Master

O tipo de sincronização master garante a corretude temporal da cadeia de dados definida como "master". Em termos dos hypercharts, trata-se de considerar o estado de origem cujo nome é colocado entre parênteses juntamente com a anotação do tipo da transição. Quando esse estado tiver sua atividade de apresentação encerrada, desde que os outros estados de origem estejam ativos, a transição irá disparar.

A figura 4.10 mostra uma versão do hyperchart da figura 4.6 em que é utilizada uma transição do tipo master, com o estado X_{12} representando a cadeia de dados "master". A transformação desse hyperchart em uma representação equivalente é apresentada na figura 4.11.

No caso das transições do tipo master, a única possibilidade de disparo da sincronização é através do componente "master", representado no exemplo pela transição entre os estados E_1 (relativo à desativação de X_{12}) e Y . O rótulo dessa transição simplesmente testa o valor de *enab* para verificar a habilitação da transição. Devido ao uso dessa variável, como ocorre com o tipo *strong-or*, são associadas atividades aos estados apropriados para que os valores corretos de *enab* sejam obtidos.

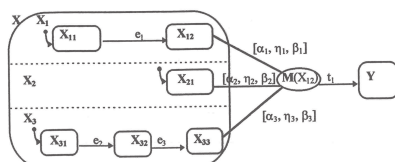
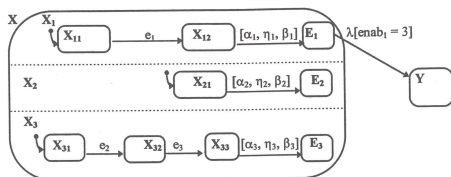


Figura 4.10 - Exemplo de transição M:N sincronizada do tipo master

O uso de uma transição associada ao evento nulo e com origem no estado que representa o final da apresentação da cadeia de dados "master" garante que os *limites temporais* relacionados com a cadeia "master" sejam respeitados. Dessa maneira, a semântica desejada para esse tipo de transição é fielmente modelada com statecharts.

Como parece claro, o uso de transições do tipo *master* é apropriado para cenários de sincronização em que a apresentação de uma determinada cadeia de dados é suficiente para a satisfação do significado global da apresentação.



O estado X recebeu a atividade
On entry enab₁ := 0
 Os estados X₁₂, X₂₁ e X₃₃ receberam, cada um, a atividade:
On entry enab₁ := cr(enab₁)+1

Figura 4.11 - Exemplo de transformação para uma transição do tipo *master*

A transformação de uma *transição M:N sincronizada* do tipo *master* em uma representação intermediária em nível semântico é expressa formalmente da seguinte forma:

Dado $t_k = (\text{Source}, \text{Target}, M, s_m)$
 $(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i]^m"), (i \in N \mid 1 \leq i \leq M) \}) \wedge$
 $(\text{Target} \subset 2^{S \times H}) \wedge \exists \langle s_m, l \rangle \in \text{Source} \mid s_m \in S$

A transformação de t_k em uma representação intermediária é dada por:

$V' = V \cup \{enab_k\}$

$C' = C \cup \{c_M\} \mid c_k = "enab_k = M"$

$L' = L \cup \{l_k\} \mid l_M = "\lambda [c_M]"$

$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

$S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$

$T' = T \cup \{t_{SEi} \mid t_{SEi} = (\{s_i\}, l_i, \{E_i\})$

se $s_i = s_m$, então $T' = T \cup \{t_{ETk} \mid t_{ETk} = (\{E_i\}, l_M, \text{Target})$

O estado s_i recebeu a atividade "**On entry** enab_k := cr(enab_k) + 1"

O estado $AncX = LCA(\{E_i \mid i \in N, 1 \leq i \leq M\})$ recebeu a atividade “On entry $enab_i := 0$ ”.

Diz-se que a representação obtida é intermediária devido ao fato de que a transformação de uma *transição M:N sincronizada* gera um modelo que ainda possui transições temporais, as quais pertencem à notação de *hypercharts* e não de *statecharts*. Em termos práticos, no entanto, considera-se que todas as transformações ocorrem em um mesmo passo e que, portanto, o nível intermediário, citado anteriormente, é eliminado.

4.4.4 Sincronização do tipo Or

A sincronização do tipo *or* introduz a noção de não-determinismo de sincronização, ou seja, é considerada a correitude temporal de qualquer uma das cadeias que terminar (e não abortar) sua *atividade de apresentação* antes das demais. Em termos dos *hypercharts*, trata-se de considerar cada *estado de origem* em termos de seu término devido ao final da *atividade de apresentação*. Além disso, a transição também deverá disparar caso todos os *estados de origem* tenham atingido seus *limites temporais máximos*. Esse tipo de transição difere do tipo *strong-or* pelo fato de permitir que alguns estados permaneçam ativos mesmo se seus *limites temporais máximos* forem atingidos. Com o tipo *strong-or*, se esse caso ocorresse, a transição dispararia imediatamente. A figura 4.12 apresenta a notação da transição *M:N sincronizada* do tipo *or*, cuja transformação para o nível semântico é apresentada na figura 4.13.

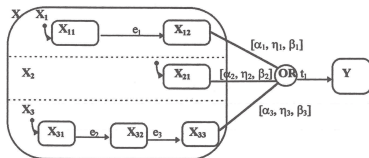
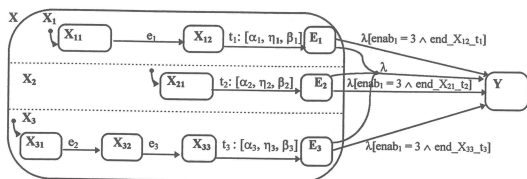


Figura 4.12 - Exemplo de transição *M:N sincronizada* do tipo *or*

A transformação realizada para transições do tipo *or* utiliza, além da variável $enab_1$, a variável que sinaliza a saída do estado de origem devido ao término da atividade de apresentação ($end_E_k_t_i$). Como visto anteriormente, essa variável resulta da transformação de uma *transição temporal*, em nível de *hyperchart*, em um *statechart* equivalente. Para clareza dos exemplos das transformações das *transições M:N sincronizadas* apresentadas neste capítulo, assume-se a existência de um nível intermediário de transformação, de modo que as transformações relativas às transições t_1 , t_2 e t_3 não são mostradas na figura. No entanto, deve-se assumir que elas existem ao nível do *statechart*.

A garantia da compatibilidade com a semântica original das transições do tipo *or* é dada, em nível da representação em *statecharts*, da seguinte forma: as transições 1:1 com origem nos estados E_k modelam o requisito de que a transição deverá disparar se todos os *estados de origem* estiverem ativos ($enab = 3$) e um desses estados tiver encerrado sua atividade de apresentação ($end_E_k_t_i = true$). A transição *M:N* convencional associada ao evento nulo modela o fato de todos os *estados de origem* serem desativados, ou por terem

encerrado suas atividades de apresentação, satisfazendo o primeiro requisito, ou por terem atingido seus limites temporais mínimos, o que satisfaz o segundo requisito. Essa combinação de restrições garante que a transição do tipo *or* disparará entre o menor dos limites temporais mínimos e o maior dos limites temporais máximos de todas as cadeias de dados envolvidas.



O estado X recebeu a atividade:
On entry enab₁ := 0
 Os estados X₁₂, X₂₁ e X₃₃ receberam, cada um, a atividade:
On entry enab₁ := cr(enab)+1

Figura 4.13 - Exemplo de transformação para uma transição do tipo *or*

Em termos formais, a transformação de uma transição *M:N sincronizada* do tipo *or* em uma representação intermediária em nível semântico é definida como segue:

Dado $f_k = (\text{Source}, \text{Target}, \text{OR}, \emptyset)$
 $(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i])", (i \in N \mid 1 \leq i \leq M) \}) \wedge$
 $(\text{Target} \subset 2^{(S \times H)})$

A transformação de f_k em uma representação intermediária é dada por:

$V' = V \cup \{enab_k\}$

$C' = C \cup \{c_{OR}\} \mid c_{OR} = "enab_k = M"$

$L' = L \cup \{l_{OR}\} \mid l_{OR} = "\lambda"$

$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

então: $S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$

$T' = T \cup \{t_{SEi}\} \mid t_{SEi} = (\{s_i\}, l_i, \{E_i\})$

$C' = C \cup \{c_{ORi}\} \mid c_{ORi} = "c_{OR} \wedge \text{end}_{s_i}_{t_{SEi}}"$

$L' = L \cup \{l_{ORi}\} \mid l_{ORi} = "\lambda[c_{ORi}]"$

$T' = T \cup \{t_{ETi}\} \mid t_{ETi} = (\{E_i\}, l_{ORi}, \text{Target})$

O estado s_i recebeu a atividade "**On entry** enab_k := cr(enab_k) + 1"

$T' = T \cup \{t_{MNR}\} \mid t_{MNR} = (\{E_i \mid i \in N, 1 \leq i \leq M\}, l_{OR}, \text{Target})$

O estado $AncX = LCA(\{E_i \mid i \in N, 1 \leq i \leq M\})$ recebeu a atividade "On entry $enab_k := 0$ ".

4.4.5 Sincronização do tipo And

O objetivo do tipo de sincronização *and* é o de garantir a correção temporal de todo o cenário de sincronização. Em termos da transição em *hypercharts*, trata-se de considerar a intersecção, se não for vazia, entre os intervalos definidos pelos *eventos temporais* associados a cada *arco de origem*.

Se o *hyperchart* da figura 4.12 contivesse uma transição *M:N sincronizada* cujo tipo fosse *and*, representado por A em vez de OR, a sua transformação semântica resultaria em uma representação intermediária como a apresentada na figura 4.14.

Uma transição do tipo *and* dispara se todos os *estados de origem* tiverem suas *atividades de apresentação* terminadas e tiverem atingido o *limite temporal mínimo*. De outra forma, se todos os estados de origem tiverem atingido seus limites temporais mínimos e ao menos um deles tiver alcançado seu *limite temporal máximo*, a transição também disparará.

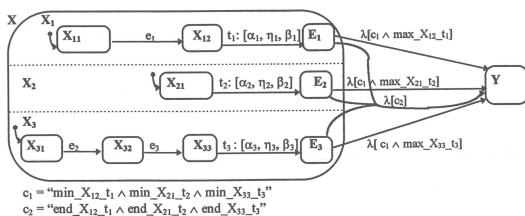


Figura 4.14 - Exemplo de transformação para uma transição do tipo *and*

Novamente, variáveis geradas pelas transformações relativas a *transições temporais* são utilizadas para a verificação da habilitação da transição (ver figuras 4.3 a 4.5). Trata-se das variáveis-padrão $\text{min_X}_{am_t_i}$. Variáveis desse tipo sinalizam que um estado de origem de uma transição temporal atingiu seu *limite temporal mínimo*. Além disso, essa variável sinaliza, indiretamente, a ativação do respectivo *estado de origem*. A combinação dos testes sobre os valores das variáveis indicadoras de fim da *atividade de apresentação* e de saída por alcance do *limite temporal máximo* faz com que o intervalo de disparo definido originalmente para transições do tipo *and* seja modelado corretamente. O uso de transições desse tipo é indicado para cenários de sincronização em que as correções temporais de todas as cadeias envolvidas sejam garantidas.

Formalmente, essa transformação é definida como segue:

Dado $t_k = (\text{Source}, \text{Target}, A, \emptyset)$

(Source = $\{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i])", (i \in N \mid 1 \leq i \leq M) \} \wedge$
 (Target $\subset 2^{S \times B}$)

A transformação de t_k em uma representação intermediária é dada por:

$$C' = C \cup c_{A1} \mid c_{A1} = " \min_{s_i} t_{SEI} \wedge \min_{s_{i+1}} t_{SEI+1} \wedge \dots \wedge \min_{s_M} t_{SEM} ", i \in N, 1 \leq i \leq M$$

$$C'' = C \cup c_{A2} \mid c_{A2} = " \text{end}_{s_i} t_{SEI} \wedge \text{end}_{s_{i+1}} t_{SEI+1} \wedge \dots \wedge \min_{s_M} t_{SEM} ", i \in N, 1 \leq i \leq M$$

$$L' = L \cup \{I_A\} \mid I_A = " \lambda[c_{A2}] "$$

($\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$)

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

então: $S' = S \cup \{E_i \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)\}$

$$T' = T \cup \{t_{SEI}\} \mid t_{SEI} = (\{s_i\}, l_i, \{E_i\})$$

$$C' = C \cup \{c_{A1}\} \mid c_{A1} = " c_{A1} \wedge \max_{s_i} t_{SEI} "$$

$$L' = L \cup \{I_A\} \mid I_A = " \lambda[c_{A1}] "$$

$$T' = T \cup \{t_{ETI}\} \mid t_{ETI} = (\{E_i\}, I_A, \text{Target})$$

O estado s_i recebeu a atividade "On entry $enab_k := cr(enab_k) + 1$ " \rightarrow ?

$$T'' = T' \cup \{t_{MNR}\} \mid t_{MNR} = (\{E_i \mid i \in N, 1 \leq i \leq M\}, I_A, \text{Target})$$

4.4.6 Sincronização do tipo Or-Master

Esse tipo de sincronização combina as semânticas de sincronização dadas pelos tipos *or* e *master*. Dessa forma, esse tipo de transição introduz a noção de não-determinismo de sincronização limitado superiormente pela restrição temporal associada à cadeia de dados "master". Assim, uma transição do tipo *or-master* disparará se um dos estados de origem (incluindo o "master") for desativado pelo término normal de sua *atividade de apresentação* ou se o estado definido como "master" alcançar seu *limite temporal máximo*.

A figura 4.15 apresenta um exemplo de transição *M:N sincronizada* do tipo *or-master*. A transformação do *hyperchart* dessa figura em uma representação equivalente é apresentada na figura 4.16.

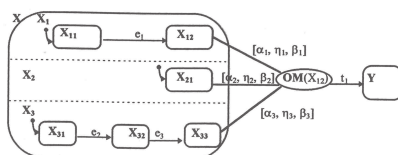


Figura 4.15 - Exemplo de transição *M:N sincronizada* do tipo *or-master*

Para a transformação relativa ao tipo *or-master* utilizou-se uma combinação do modelo gerado para o tipo *or* com aquele gerado para o tipo *master*. Assim, o uso de uma variável do

nível de nível de statechart e a criação da variável *enab* permitem a checagem das condições de disparo. A definição formal da transformação de uma transição do tipo *or-master* em uma representação intermediária é dada a seguir:

Dado $t_k = (\text{Source}, \text{Target}, \text{OM}, s_m)$

$$(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i]", (i \in N \mid 1 \leq i \leq M)) \} \wedge (\text{Target} \subset 2^{(S \times I^O)}) \wedge \exists \langle s_m, l \rangle \in \text{Source} \mid s_m \in S$$

A transformação de t_k em uma representação intermediária é dada por:

$$V' = V \cup \{ \text{enab}_k \}$$

$$C' = C \cup c_{\text{OM}} \mid c_{\text{OM}} = \text{"enab}_k = M"$$

$$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$$

Seja $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

$$S' = S \cup \{ E_i \} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$$

$$T' = T \cup \{ t_{\text{SEI}} \} \mid t_{\text{SEI}} = (\{ s_i \}, l_i, \{ E_i \})$$

se $s_i = s_m$, então: ($L' = L \cup \{ l_{m_k} \} \mid l_{m_k} = \text{"}\lambda[c_{1k}] \text{"}$)

$$T' = T \cup \{ t_{\text{ETMI}} \} \mid t_{\text{ETMI}} = (\{ E_i \}, l_{m_k}, \text{Target})$$

senão: ($C' = C \cup \{ c_{\text{OM}_i} \} \mid c_{\text{OM}_i} = \text{"}c_{\text{OM}} \wedge \text{end_}s_i \text{"}$)

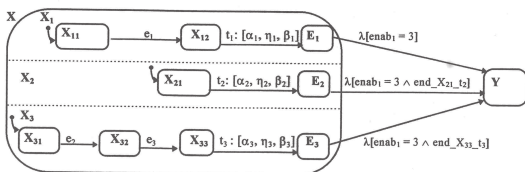
$$L' = L \cup \{ l_{\text{OM}_i} \} \mid l_{\text{OM}_i} = \text{"}\lambda[c_{\text{OM}_i}] \text{"}$$

$$T' = T \cup \{ t_{\text{ETI}} \} \mid t_{\text{ETI}} = (\{ E_i \}, l_{\text{OM}_i}, \text{Target})$$

O estado s_i recebeu a atividade "**On entry** $\text{enab}_k := \text{cr}(\text{enab}_k) + 1$ ")

O estado $\text{AncX} = \text{LCA}(\{ E_i \mid i \in N, 1 \leq i \leq M \})$ recebeu a atividade "**On entry** $\text{enab}_k :=$

0".



O estado X recebeu a atividade:

On entry $\text{enab}_1 := 0$

Os estados X_{12} , X_{21} e X_{33} receberam, cada um, a atividade:

On entry $\text{enab}_1 := \text{cr}(\text{enab}_1) + 1$

Figura 4.16 - Exemplo de transformação para uma transição do tipo *or-master*

4.4.7 Sincronização do tipo *Strong-Master*

A sincronização do tipo *strong-master* é o resultado da combinação dos tipos *strong-or* e *master*. As restrições temporais consideradas por esse tipo de sincronização são aquelas relativas à cadeia de dados "master" e à cadeia de dados mais avançada. Em termos de *hypercharts*, isso significa que a transição irá disparar se o estado "master" tiver atingido o seu *limite temporal mínimo* e algum dos estados (incluindo o "master") tiver sido desativado, seja pelo término da sua *atividade de apresentação* ou por ter atingido seu *limite temporal máximo*.

Se o símbolo OM utilizado na transição *M:N sincronizada* do hyperchart apresentado na figura 4.15 fosse trocado pelo símbolo SM, representando o tipo *strong-master*, a sua transformação resultaria em um modelo como o da figura 4.17.

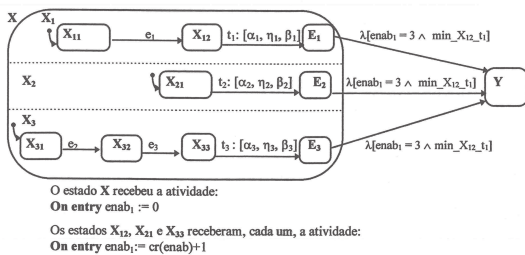


Figura 4.17 - Exemplo de transformação para uma transição do tipo *strong-master*

Formalmente, essa transformação é definida como segue:

Dado $t_k = (\text{Source}, \text{Target}, SM, s_m) |$

$$(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i]"), (i \in N \mid 1 \leq i \leq M) \}) \wedge (\text{Target} \subset 2^{(S \times H)}) \wedge \exists \langle s_m, l \rangle \in \text{Source} \mid s_m \in S$$

A transformação de t_k em uma representação intermediária é dada por:

$$V' = V \cup \{enab_k\}$$

$$C' = C \cup c_{SM} \mid c_{SM} = "enab_k = M \wedge \min_s_m_t_{ETM}"$$

$$L' = L \cup \{l_{SM}\} \mid l_{SM} = "\lambda[c_{SM}]"$$

$$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$$

Seja $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

$$S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$$

$$T = T \cup \{t_{SEI}, t_{ETM}\} \mid t_{SEI} = (\{s_i\}, l_i, \{E_i\}) \wedge t_{ETM} = (\{E_i\}, l_{SM}, Target)$$

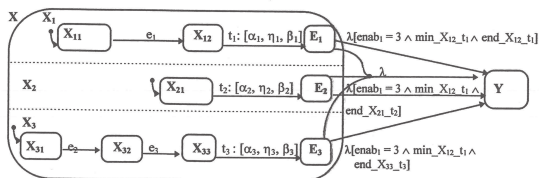
O estado s_i recebeu a atividade "On entry $enab_k := cr(enab_k) + 1$ "

O estado $AncX = LCA(\{E_i \mid i \in N, 1 \leq i \leq M\})$ recebeu a atividade "On entry $enab_k := 0$ ".

4.4.8 Sincronização do tipo Weak-Master

O tipo de sincronização *weak-master* combina as semânticas de sincronização dadas pelos tipos *weak-and* e *master*. Esse tipo de transição é complementar ao tipo *or-master*, introduzindo a noção de não-determinismo de sincronização limitado inferiormente pela restrição temporal associada à cadeia de dados "master". Assim, em termos de *hypercharts*, uma transição do tipo *weak-master* disparará se o estado de origem considerado "master" tiver atingido seu limite temporal mínimo e um dos estados de origem (incluindo o "master") tiver sido desativado pelo término normal de sua atividade de apresentação (e não por ter atingido seu limite temporal máximo). Além disso, se todos os estados de origem forem desativados por terem atingido seus limites temporais máximos a transição também deverá disparar.

Assim, a representação intermediária do *hyperchart* da figura 4.15, caso a transição seja do tipo *weak-master*, o que é obtido substituindo o símbolo OM por WM, é apresentada na figura 4.18.



O estado X recebeu a atividade:

On entry $enab_1 := 0$

Os estados X_{12} , X_{21} e X_{33} receberam, cada um, a atividade:

On entry $enab_1 := cr(enab)+1$

Figura 4.18 - Exemplo de transformação para uma transição do tipo *weak-master*

A transformação de uma transição $M:N$ sincronizada do tipo *weak-master* em uma representação intermediária em nível semântico é formalizada da seguinte forma:

Dado $t_k = (Source, Target, WM, s_m)$

$$(Source = \{<s_i, l_i> \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i]"), (i \in N \mid 1 \leq i \leq M)\}) \wedge (Target \subset 2^{S \times H}) \wedge \exists <s_m, l_m> \in Source \mid s_m \in S$$

A transformação de t_k em uma representação intermediária é dada por:

$$V' = V \cup \{enab_k\}$$

$$C' = C \cup \{c_{WM}\} \mid c_{WM} = "enab_k = M \wedge \min_s_m_t_{SEM}"$$

$$L' = L \cup \{l_{WM}\} \mid l_{WM} = "\lambda"$$

$$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M)$$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

$$S' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$$

$$T' = T \cup \{t_{SEi}\} \mid t_{SEi} = (\{s_i, l_i, \{E_i\}\})$$

(se $s_i = s_m$, então: $(t_{SEM} = t_{SEi})$)

$$C'' = C \cup \{c_{WMI}\} \mid c_{WMI} = "c_{WM} \wedge \text{end_}s_i_t_{SEi}"$$

$$L' = L \cup \{l_{WMI}\} \mid l_{WMI} = "\lambda, [c_{WMI}]"$$

$$T' = T \cup \{t_{ETi}\} \mid t_{ETi} = (\{E_i, l_{WMI}, \text{Target}\})$$

O estado s_i recebeu a atividade "On entry $enab_k := cr(enab_k) + 1$ "

$$T' = T \cup \{t_{MNR}\} \mid t_{MNR} = (\{E_i \mid i \in N, 1 \leq i \leq M\}, l_{WM}, \text{Target})$$

O estado $AncX = LCA(\{E_i \mid i \in N, 1 \leq i \leq M\})$ recebeu a atividade "On entry $enab_k := 0$ ".

4.4.9 Sincronização do tipo And-Master

Caso a transição apresentada na figura 4.15 tenha o símbolo AM associado, em vez de OM, ou seja, se essa transição for do tipo and-master, o modelo intermediário equivalente àquele hyperchart terá a forma apresentada na figura 4.19.

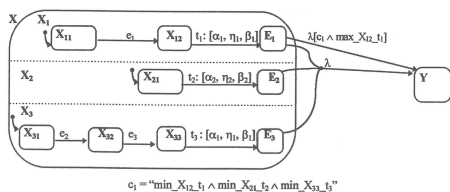


Figura 4.19 - Exemplo de transformação para uma transição do tipo and-master

Esse tipo de sincronização é uma combinação das semânticas dadas pelos tipos and e master. Assim, o limite superior do intervalo dinâmico de disparo da transição fica determinado pela cadeia "master", enquanto que o limite inferior fica determinado pela cadeia mais atrasada. Dessa forma, o disparo de uma transição desse tipo irá ocorrer nas seguintes situações: se todos os estados de origem tiverem atingido os seus limites temporais mínimos e se o estado "master" atingir seu limite temporal máximo; se todos os estados de origem terminarem suas

atividades de apresentação. Certamente, se todos os estados atingirem os seus limites temporais máximos, a condição irá valer para o estado "master", fazendo com que a transição desapareça.

A formalização da transformação de uma transição M:N sincronizada do tipo and-master em uma representação intermediária é definida como segue:

Dado $t_k = (\text{Source}, \text{Target}, A.M, s_m)$

$(\text{Source} = \{ \langle s_i, l_i \rangle \mid s_i \in S, (l_i \in L \mid l_i = "[\alpha_i, \eta_i, \beta_i])", (i \in N \mid 1 \leq i \leq M) \})$
 $\wedge (\text{Target} \subset 2^{S \times B}) \wedge \exists \langle s_m, l \rangle \in \text{Source} \mid s_m \in S$

A transformação de t_k em uma representação intermediária é dada por:

$C' = C \cup \{c_{AM}\} \mid c_{AM} = \text{"max_s_m_t_{SEM} \wedge \min_{s_i} t_{SEi} \wedge \min_{s_{i+1}} t_{SEi+1} \wedge \dots \wedge \min_{s_M} t_{SEM}"}$, $i \in N, 1 \leq i \leq M$

$L' = L \cup \{l_{AM1}\} \mid l_{AM1} = \text{"}\lambda[c_{AM}]\text{"}$

$L' = L \cup \{l_{AM2}\} \mid l_{AM2} = \text{"}\lambda\text{"}$

$(\forall \langle s_i, l_i \rangle \in \text{Source} \mid i \in N, 1 \leq i \leq M$

Sejam $X_i \in S \mid s_i \in \rho(X_i)$ e um novo estado E_i

$S'' = S \cup \{E_i\} \mid \rho(E_i) = \emptyset \wedge E_i \in \rho(X_i)$

$T' = T \cup \{t_{SEi}\} \mid t_{SEi} = (\{s_i, l_i\}, \{E_i\})$

se $s_i = s_m$ então $(T' = T \cup \{t_{ETm}\} \mid t_{ETm} = (\{E_i\}, l_{AM1}, \text{Target})$

$t_{SEM} = t_{SEi}$

$T' = T \cup \{t_{MNR}\} \mid t_{MNR} = (\{E_i \mid i \in N, 1 \leq i \leq M\}, l_{AM2}, \text{Target})$

4.5 Parametrização de estados e abstração de transições

A modelagem de aplicações hipermédia demanda grande variação de níveis de abstração e de granularidade da especificação, gerando a necessidade de que o modelo de especificação suporte esses mecanismos de forma conveniente. Além disso, em aplicações de multimídia, há a repetição de certos padrões de comportamento. Um exemplo disso é o caso de uma aplicação que necessite de sincronização quadro-a-quadro entre áudio e vídeo. O padrão repetitivo nesse caso é a especificação dos dois quadros, um de vídeo e um de áudio e a sincronização aplicada a eles, levando a um outro cenário de sincronização do mesmo tipo. Considerando o uso da codificação comum de vídeo digital, essa especificação deveria tratar trinta quadros por segundo de apresentação. Isso quer dizer que, para uma apresentação de dez segundos, por exemplo, seriam necessárias trezentas repetições do mesmo cenário de sincronização, o que tornaria a especificação demasiadamente grande.

Assim, seria interessante, para tornar a notação mais concisa e econômica, que tais padrões repetitivos pudessem ser especificados uma só vez, acompanhados da especificação formal de quantas vezes se repetem.

A solução adotada pelos *hypercharts* é derivada da notação de *estados parametrizados*, sugeridos por Harel em [Har87b], além de um novo conceito, o de *abstração*

de transições, que permite que um estado seja decomposto isoladamente, independentemente das transições que cruzam suas bordas, chegando aos seus subestados ou partindo deles. A notação proposta é apresentada a seguir.

4.5.1 Estados parametrizados

A notação de *estados parametrizados*, como sugerida por Harel, permite que estados e respectivas transições com comportamentos iguais sejam representados uma única vez através de um estado e um conjunto de transições acompanhados de uma notação de indexação, a qual determina um *intervalo de parametrização* sobre o qual são definidos os estados. Assim, se um *intervalo de parametrização* $[1,10]$ é definido, isso é equivalente a se escrever dez estados e um conjunto de transições correspondente em uma notação não-parametrizada. Como a proposta de parametrização de Harel não é completamente definida, a notação foi modificada de forma a aumentar o poder de especificação dos *hypercharts* e, conseqüentemente, do modelo XHMBS.

Diferentemente das outras notações apresentadas nas seções anteriores, a *parametrização de estados*, quando expandida, possui um *hyperchart* equivalente ao invés de um statechart. Dessa forma, a transformação semântica de uma especificação parametrizada deve ser feita em dois passos: a expansão do estado parametrizado e a transformação dos *hypercharts* resultantes em seus statecharts equivalentes em nível semântico. A expansão das notações parametrizadas não é formalizada neste trabalho, sendo mostrada somente através de exemplos.

Em *hypercharts*, como em statecharts, a parametrização é sempre relativa a um superestado cujos subestados são descritos pela notação parametrizada. No entanto, diferentemente da proposta de Harel, *hypercharts* permitem parametrizações de estados do tipo AND e do tipo OR. A notação para ambas as parametrizações é bastante semelhante, diferindo pelo uso dos símbolos matemáticos dos operadores *or* (\vee), para decomposições exclusivas, e *and* (\wedge) para decomposições ortogonais.

A declaração da forma de parametrização de um estado inclui, além do seu tipo (*or* ou *and*), o nome do superestado, o nome genérico dos subestados que serão parametrizados, a *variável de parametrização* (ou *índice*) e a definição do *intervalo de parametrização*, cujo domínio é sempre o conjunto dos números naturais. A figura 4.20-a apresenta um exemplo de parametrização simples do tipo OR. O conjunto de subestados, representado graficamente por um agrupamento em perspectiva é o objeto da parametrização. Suas instâncias são determinadas pelo *índice*, que varia dentro do intervalo definido. A figura 4.20-b mostra como seria o *hyperchart* equivalente em notação expandida. Esse exemplo não contém transições, cuja especificação parametrizada é apresentada adiante.

A parametrização simples do tipo AND, assim como o *hyperchart* equivalente são mostrados na figura 4.21.

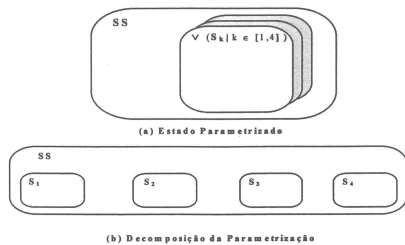


Figura 4.20 - Parametrização simples do tipo OR

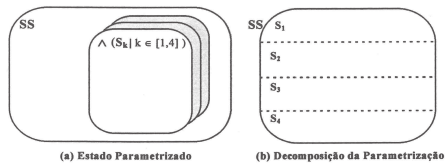


Figura 4.21 - Parametrização simples do tipo AND

A possibilidade de a parametrização envolver transições associadas aos subestados parametrizados gera a necessidade de uma notação apropriada que permita especificar qual é exatamente o estado envolvido na origem ou no destino de uma determinada transição. Em outras palavras, é necessário especificar o valor da *variável de parametrização* que está sendo referenciado pela transição. Na maioria dos casos, porém, a identificação de um estado só pode ser especificada de forma relativa, ou seja, não é possível determinar o valor exato da *variável de parametrização*, mas esse valor pode ser determinado se o valor da variável que determina o estado anterior (ou posterior, conforme o problema) for conhecido. Certamente, esse processo de valorização é recursivo e deve satisfazer às restrições determinadas pelo *intervalo de parametrização*.

Em termos da notação em *hypercharts*, utiliza-se o mesmo símbolo sugerido em [Har87b], ou seja, um losango contendo uma sentença que define o valor da *variável de parametrização*. Essa sentença, aqui chamada de *sentença de valorização*, é uma expressão aritmética que determina o valor da *variável de parametrização*. Esse valor pode ser resultado de um cálculo relativo ao *índice* do estado ou uma constante. A figura 4.22 apresenta o exemplo da figura 4.20 com transições associadas à parametrização. Deve-se notar que, quando a *sentença de valorização* resulta em um valor que não estiver definido no *intervalo de parametrização*, a transição associada não é definida.

No exemplo da figura 4.22-a, as *sentenças de valoração* são utilizadas para três definições de parametrização. A primeira delas (“ $k=1$ ”) define que o estado S_1 é o subestado “default” do estado SS . As outras duas dizem respeito às transições entre os estados, uma associada ao evento e_1 , e outra associada ao evento e_2 . Como mostra a figura 4.22-b, quando expandida, essa parametrização possui duas exceções: a ausência de uma transição associada ao evento e_2 partindo do estado S_1 e de uma transição associada ao evento e_1 partindo de S_4 . Isso se deve ao fato de que o valor da *variável de parametrização* não está definido, não sendo possível definir o estado de destino dessas transições, o que as torna igualmente indefinidas.

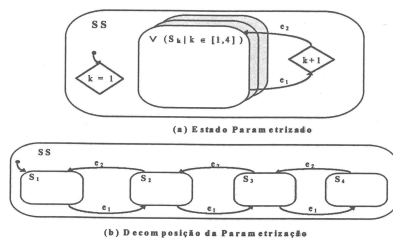


Figura 4.22 - Parametrização do tipo OR com transições associadas

É comum, no entanto, que se deseje definir exceções à parametrização, principalmente para o primeiro e o último estados de uma seqüência encadeada por transições entre eles. Casos desse tipo ocorrem quando se deseja especificar transições entre um estado contido na parametrização e um outro externo a ela. A solução para esse problema é dada em *hypercharts* através da *reescrita de comportamento* que, em termos notacionais, é a definição de uma *sentença de valoração* com valor constante, representando explicitamente um estado que pode ser origem ou destino de transições associadas a estados externos à notação parametrizada. A figura 4.23 apresenta uma modificação do exemplo da figura 4.20 em que essa notação é utilizada.

Apesar do nome “reescrita de comportamento” dar a idéia de redefinição, a semântica dessa notação simplesmente adiciona elementos à especificação da parametrização. Assim, como pode ser visto na figura 4.23-b, o estado S_4 , que foi reescrito, possui todas as características definidas nos outros termos da parametrização além de ser origem da transição definida através da reescrita de seu comportamento.

O uso de *sentenças de valoração* é menos útil em parametrizações do tipo AND, uma vez que não é permitido aos subestados parametrizados, que são concorrentes, possuírem transições entre eles. No entanto, podem ser especificadas parametrizações do tipo OR com subestados parametrizados do tipo AND, como mostra a figura 4.24. Com o uso de *abstrações de transições*, apresentadas a seguir, surgem novas utilizações para as sentenças de valoração e reescrita de comportamento, além de extensões a essa notação, como dito anteriormente.

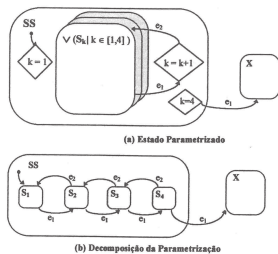


Figura 4.23 - Parametrização do tipo OR com reescrita de comportamento

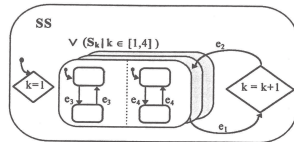


Figura 4.24 - Parametrização do tipo OR com subestados do tipo AND

4.5.2 Abstração de transições

Em [Har87b] é proposta uma notação para transições intermíveis em statecharts através da qual, tendo sido definido um nível de abstração, é possível representar apenas o contexto em que determinada transição se encontra, sem que seja necessário explicitar seus estados de origem e de destino. Trata-se de um mecanismo que abstrai transições em termos de superestados de origem e de destino, como mostrado na figura 4.25. Nesse exemplo a transição t_2 , associada ao evento b , tem origem no estado Y e destino no estado X_1 . Assim, quando a mesma especificação é vista em um nível mais alto de abstração (figura 4.25-b), a notação de *abstração de transições* é utilizada e o destino de t_2 é representado apenas pelo superestado X , o que significa que o estado de destino dessa transição é um subestado de X . O mesmo ocorre com a transição t_3 , cujo estado de origem é X_2 . Quando o nível de abstração é aumentado, a notação é utilizada para indicar que a origem de t_3 é um subestado de X . A transição t_1 permanece com a forma original, visto que está definida no mesmo nível de abstração em que se encontram os estados X e Y .

O componente notacional utilizado para suportar o mecanismo de *abstração de transições* é o *ponto de conexão*, representado por um corte de perfuração na superfície do estado, denotando que a transição atravessa a sua borda.

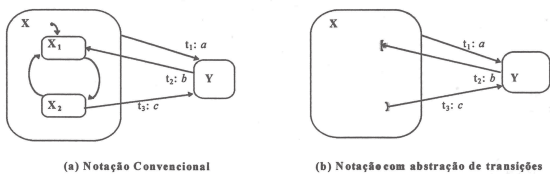


Figura 4.25 - Abstração de transições

Essa abordagem pode ser utilizada tanto para a especificação de relações entre superestados determinadas por transições entre seus subestados, quanto para a decomposição independente desses superestados. Dessa forma, utilizando o mecanismo de *abstração de transições*, torna-se possível realizar a especificação da estrutura de um estado sem que seja necessário explicitar as suas relações — via transições — com outros estados do modelo, como mostra a figura 4.26.

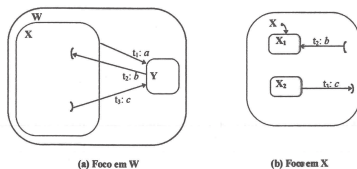


Figura 4.26 - Uso de abstração de transições para expansão de estados

Nesse exemplo, o uso de um identificador de transição (t_i) permite especificar explicitamente qual é a transição especificada no nível mais alto (4.26-a) e a *transição abstrata* correspondente no nível expandido (4.26-b). Podem existir casos, no entanto, em que não seja conveniente utilizar identificadores de transição, como ocorre com estados parametrizados. Nesses casos, os *pontos de conexão* devem ser rotulados e a correspondência deve ser especificada através deles nos vários níveis de abstração que forem abordados na modelagem.

A figura 4.27 é uma variação do exemplo apresentado na figura 4.26 em que os *pontos de conexão* são rotulados. Por convenção os rótulos são sempre colocados dentro de um pequeno círculo. No caso da figura 4.27, os eventos são identificados somente no nível mais alto de abstração (4.27-a). Uma especificação poderia adotar o procedimento contrário, especificando os eventos nos níveis mais baixos de abstração. Uma possibilidade bastante interessante, para a qual o desenvolvedor deve ser encorajado, é a de se especificar os eventos no nível de abstração em que é especificado o estado de origem da transição. No entanto, desde que os eventos sejam especificados em alguma parte do documento, a abordagem de especificação deverá ser aquela com a qual o desenvolvedor se sentir mais à vontade.

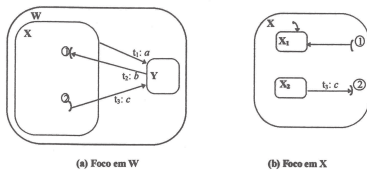


Figura 4.27 - Pontos de conexão rotulados

4.5.3 Estados parametrizados com abstração de transições

O uso combinado dos mecanismos de *parametrização de estados* e *abstração de transições* constitui uma técnica de especificação bastante poderosa, permitindo especificar estruturas repetitivas complexas através de diagramas bastante concisos. A abordagem adotada é estruturada, baseada em blocos de construção definidos pelo desenvolvedor. Basicamente, pode-se utilizar, de forma combinada, *pontos de conexão* rotulados associados às *sentenças de valoração* e ao *estado parametrizado*, definindo a correspondência entre as transições definidas no nível de parametrização e aquelas definidas nos níveis mais baixos de abstração. Um exemplo dessa abordagem é apresentado na figura 4.28, na qual a parametrização do estado é especificada e as transições correspondentes são abstraídas. Os subestados parametrizados são especificados no nível imediatamente inferior de abstração, mostrado na figura 4.28-b.

No exemplo da figura 4.28 os *pontos de conexão* e as *sentenças de valoração* (dentro dos losangos) são utilizados em conjunto especificando que a transição associada ao evento ev_1 tem origem e destino em subestados S_k . O *ponto de conexão* dentro do losango indica que a transição chega naquele *ponto de conexão* do estado especificado pela *sentença de valoração*. A decomposição dos estados S_k , apresentada na figura 4.28-b, mostra quais são os subestados que participam da transição abstraída em 4.28-a. Assim, com as informações especificadas nos dois níveis de abstração, sabe-se que a transição associada ao evento ev_1 parte de um subestado $S_k.Sub_{k2}$ e chega em um estado $S_{(k+1).Sub_{(k+1)1}}$. Esse exemplo é apresentado na figura 4.29 com notação expandida.

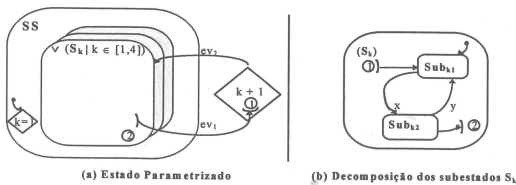


Figura 4.28 - Combinação de parametrização com abstração

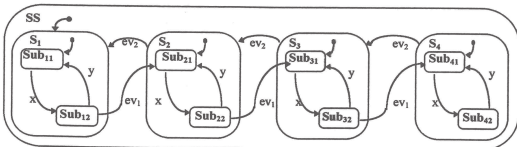


Figura 4.29- Expansão de parametrização com transições abstratas

É importante notar que, mesmo sem haver qualquer ponto de conexão "1" no estado parametrizado da figura 4.28-a, o uso de um identificador "1" junto à sentença de valoração "k+1", especificando-o como destino da transição associada ao evento ev_1 , quando unido à expansão de S_k na figura 4.28-b, permite que a transição seja definida completamente.

Outro recurso bastante poderoso da notação de *abstração de transições* é a utilização de ligações entre *pontos de conexão*, ou seja, uma *transição* é abstraída tanto em termos de um superestado de origem quanto um superestado de destino. As figuras 4.30 a 4.33 ilustram a utilização desse mecanismo.

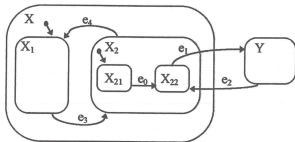
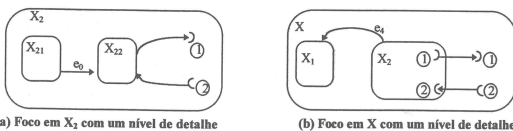


Figura 4.30 - Um exemplo com todos os níveis de abstração expandidos

Na figura 4.30, um *hyperchart* é especificado normalmente, sem o uso de abstrações de transições. A figuras 4.31-a e 4.31-b apresentam essa mesma especificação, respectivamente, com os focos nos estados X_2 e X .



(a) Foco em X_2 com um nível de detalhe (b) Foco em X com um nível de detalhe

Figura 4.31 - Especificações alternativas com uso de abstração de transições

O que ocorre com a especificação apresentada na figura 4.31 é que devido à existência de três níveis de hierarquia envolvendo as transições, elas não podem ter seus estados de origem ou de destino especificados explicitamente quando o foco da especificação é voltado para o estado X. Assim, a origem e o destino das transições são especificados em termos de *pontos de conexão* e podem ser vistos quando o foco da especificação é voltado para um nível mais alto de abstração: aquele que envolve os estados X e Y como decomposição de algum superestado. A figura 4.32 apresenta a visão da especificação nesse nível.

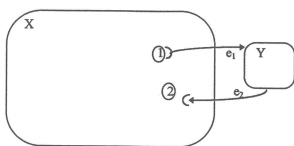


Figura 4.32 - Especificação com foco acima dos estados X e Y

Finalmente, na figura 4.33, a especificação apresentada na figura 4.30 é apresentada juntamente com todos os *pontos de conexão* utilizados, ilustrando o funcionamento dessa abordagem de especificação.

Como essa abordagem sempre coloca o foco da especificação em um único nível de abstração, as transições abstraídas em termos de suas origens e destinos sempre serão delimitadas por dois pontos de conexão em cada nível, independentemente do número de níveis do *hyperchart* que elas cruzarem. Somente no nível de abstração em que o estado de origem ou de destino de uma transição for especificado um dos pontos deixará de ser necessário.

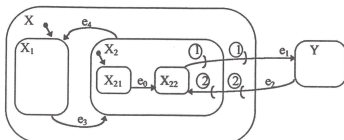


Figura 4.33 - Utilização de abstração de transições com vários níveis de hierarquia

4.5.4 Parametrização com notação temporal, mecanismos de sincronização e abstração de transições

As soluções em nível de notação providas pelos *hypercharts* visam, principalmente, a especificação temporal em um nível mais alto do que aquele oferecido por *statecharts*. Os mecanismos de *parametrização de estados* e de *abstração de transições*, no entanto, não constituem, por eles sós, facilidades para esse tipo de especificação. É possível, porém, que a

combinação de todos os recursos notacionais oferecidos por *hypercharts*, incluindo *parametrização de estados* e *abstração de transições*, constitui uma forma poderosa de especificação de cenários complexos de sincronização hipermédia, satisfazendo aos requisitos de expressividade exigidos de um modelo com esse propósito.

A abordagem adotada para a especificação em XHMBS especifica rótulos de transições nos níveis de abstração em que são especificados os seus estados de origem. Além disso, quando for conveniente, para maior clareza da especificação, esse rótulo poderá ser especificado novamente em outros níveis de abstração. O exemplo apresentado na figura 4.34 mostra a especificação de um cenário de sincronização bastante complexo, cujo tamanho da especificação expandida equivalente seria inaceitável até mesmo com apoio computadorizado. Um detalhe deve ser notado na expansão do estado S_k : o uso dos subestados a_k e v_k como subestados de $Audio_k$ e $Video_k$, respectivamente. Esses subestados seriam desnecessários em statecharts convencionais, uma vez que cada um deles é o único filho de seu respectivo estado-pai. No entanto, devido à forma com que as transformações de *transições M:N sincronizadas* são especificadas, deve-se adotar essa forma de especificação sempre que uma das cadeias a do cenário de sincronização for representada por apenas um estado.

O exemplo especifica um trecho de áudio sincronizado a cada um terço de segundo (333 milissegundos) com um trecho de vídeo. Esse cenário tem dez segundos de duração e, portanto, considerando que para cada segundo de apresentação devem ser especificados três subcenários contendo um quadro de áudio e um de vídeo sincronizados ao final de 333 milissegundos, a especificação envolveria trinta blocos desse tipo.

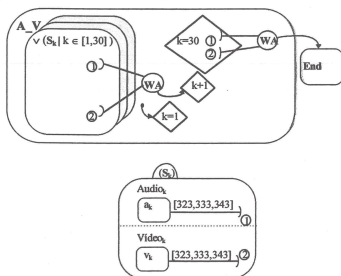


Figura 4.34 - Combinação das notações de *hypercharts*

A abordagem do modelo HMBS e também do modelo XHMBS, define cada quadro de áudio e de vídeo como uma página e, portanto, o statechart (ou o *hyperchart*) deverá possuir um estado específico para representar o comportamento de cada página no hiperdocumento. A ideia permanece a mesma com o modelo XHMBS. Assim, em termos de *hypercharts*, cada subcenário de sincronização é representado por um estado do tipo AND cujos subestados representam as cadeias de áudio e de vídeo. Além disso, deve haver uma *transição M:N*

sincronizada que leva o sistema desses subestados aos próximos, especificados de forma similar. A notação parametrizada diminui sensivelmente o número de símbolos utilizados pela especificação sem, no entanto, perder em objetividade. A figura 4.35 apresenta, informalmente, a expansão da especificação desse exemplo.

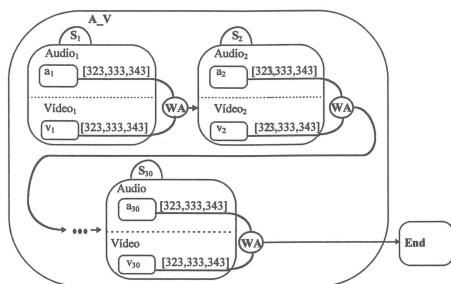


Figura 4.35 - Expansão informal de uma especificação parametrizada complexa

4.6 Considerações finais

Os *hypercharts* oferecem um conjunto adicional de notações que permite a especificação comportamental de hiperdocumentos baseada em statecharts. O uso dessa técnica permite a especificação de requisitos temporais e de sincronização relativamente complexos sem que o especificador tenha que criar novas estruturas como blocos de construção. Na verdade, a abordagem baseada em blocos de construção para o caso da especificação baseada em *hypercharts* fica oculta, em grande parte, no nível semântico da técnica, em que as notações de *hypercharts* são transformadas em statecharts convencionais. Com isso, o modelo XHMBS, que utiliza os *hypercharts* como técnica subjacente, torna-se, ao mesmo tempo, mais simples e poderoso, permitindo que especificações de grandes hiperdocumentos sejam realizadas de forma concisa, sem perda de legibilidade. Além disso, a definição de blocos de construção pode ser adicionada ao modelo em forma de diretrizes de especificação, ou seja, pode-se sugerir ao especificador o uso de certas estruturas de estados e transições, de forma que os documentos possuam maior legibilidade e padronização. Algumas dessas diretrizes são apresentadas no capítulo 5.

A semântica dos *hypercharts*, definida sobre statecharts convencionais, cuja semântica é baseada em um modelo discreto de tempo (caracterizando os passos), possui algumas implicações que podem ser consideradas relevantes quando da implementação de uma especificação feita com XHMBS. O uso de estados intermediários e eventos de "time-out" insere, por conveniência de especificação, atrasos de um ou dois passos na modelagem das transições temporais e das transições M:N sincronizadas. Por exemplo, no modelo em nível de statechart da figura 4.3, para uma transição temporal t_1 , observa-se que, se o evento de

término da atividade de apresentação ocorrer quando a aplicação estiver no subestado X_{t_1} do TCS no momento exato em que o limite temporal mínimo for atingido, a transição t_1 irá disparar somente no próximo passo. Isso porque ainda será necessário que a aplicação passe pelo estado $Wait_{X_{t_1}_{min}}$, inserido pela modelagem, para aguardar um evento de “time-out”, o qual, nesse caso específico, será associado a uma espera de zero passo, como se fosse um evento nulo.

Algo parecido ocorre com as transformações de transições $M:N$ sincronizadas, nas quais são definidas transições associadas ao evento nulo em um estado intermediário (E_i) inserido entre os estados de origem e de destino daquelas transições. Isso faz com que um atraso de um passo seja sempre inserido na modelagem. Combinados, esses atrasos podem se acumular e alterar, mesmo que sutilmente, as propriedades temporais da aplicação sob especificação. Possíveis alternativas para uma definição da semântica que evite esse problema são o uso de estados instantâneos no nível semântico, ou mesmo uma semântica baseada em timed-statecharts, cujo modelo de tempo é contínuo e cujas propriedades semânticas permitem outras formas para modelar a passagem do tempo. Para o problema relacionado com a modelagem semântica das transições temporais, a solução mais simples é a inserção de uma nova transição que tenha origem no subestado X_{t_1} do TCS e destino em Y . A expressão de eventos dessa transição seria “end $X \wedge tm(X_{t_1}, \alpha) / a$ ”, considerando, portanto, a ocorrência simultânea do evento de término da atividade de apresentação do estado X e o alcance do limite temporal mínimo de t_1 .

A função da semântica apresentada nesse capítulo é a de caracterizar o funcionamento de uma nova notação, especialmente criada para a especificação de hiperdocumentos, cuja implementação seja compatível com semânticas conhecidas de statecharts. O fato de a semântica dos *hypercharts* ser baseada em statecharts convencionais permite, portanto, que sua implementação seja facilitada, por exemplo, sobre o ambiente Statsim [Mas91], o que constitui um ponto inicial para a definição de uma máquina de *hypercharts* para um ambiente de apoio ao modelo XHMBS, como discutido no próximo capítulo.

Finalmente, o complemento à notação de *hypercharts* é dado pela definição dos componentes do modelo XHMBS. A união da notação temporal dos *hypercharts* com as definições e mecanismos de especificação do XHMBS deve constituir um conjunto notacional poderoso o suficiente para satisfazer os requisitos de aplicações hipermídia classificados no capítulo 2 que não podem ser satisfeitos pelo modelo HMBS. A definição do modelo XHMBS é dada no capítulo seguinte.

Capítulo 5

Especificação do modelo XHMBS

5.1 Um modelo para documentos de hipermídia

O XHMBS (eXtended Hyperdocument Model Based on Statecharts) é uma extensão do modelo HMBS cuja principal característica é a capacidade de especificação de requisitos relativos à sincronização em hiperdocumentos. Além disso, outros requisitos relativos às aplicações hipermídia passam a ser passíveis de especificação pelo XHMBS devido às inovações incorporadas a esse modelo. Entre as inovações apresentadas pelo modelo XHMBS estão a possibilidade de especificação da coordenação espacial e da distribuição da informação através de *canais de apresentação e comunicação*, a separação total entre o conteúdo da informação e a sua estrutura através do uso melhorado de *âncoras* e, como modificação de maior impacto, a especificação navegacional através de *hypercharts*, que passam a substituir os statecharts convencionais como técnica subjacente ao modelo.

O XHMBS continua a ser, como o HMBS, um modelo de projeto de hiperdocumentos. Dessa forma, para se especificar uma aplicação de forma completa, há a necessidade de um processo mais abrangente, partindo da modelagem semântica da informação e incluindo o XHMBS como modelo da estrutura e da semântica de navegação de um hiperdocumento. A mesma necessidade passa a existir se a aplicação envolver instâncias de hiperdocumentos com estrutura e comportamento idênticos. Nesse caso, alguma notação de mais alto nível deve existir que permita a especificação de como as instâncias dos hiperdocumentos serão criadas e de como elas se relacionarão.

Diante dessas características, a definição do modelo XHMBS, assim como a do seu antecessor, o HMBS, é baseada na definição de um hiperdocumento. Nas seções seguintes o hiperdocumento do modelo XHMBS é definido e suas novas características são apresentadas. Para exemplificar o uso do XHMBS, um exemplo conhecido de documento multimídia é especificado e os resultados dessa especificação são comparados com os resultados obtidos com especificações realizadas através de outros modelos. Além disso, o XHMBS é confrontado com os requisitos de aplicações hipermídia apresentados no capítulo 2.

5.2 O hiperdocumento XHMBS

Um hiperdocumento, segundo o XHMBS, é uma extensão do hipertexto definido pelo HMBS. As idéias básicas de estruturação e semântica de navegação do HMBS foram mantidas, de modo que a filosofia de especificação permaneceu a mesma. Assim, um hiperdocumento definido pelo XHMBS é definido como segue:

Seja *Info* o conjunto das porções básicas do conteúdo da informação de um hiperdocumento. Dessa forma, um hiperdocumento é definido como uma 11-tupla $H = \langle Hyp, P, Anc, M, A, L, Pch, Cch, V_p, V_n, Ch \rangle$ em que:

- **Hyp** é um *hyperchart* definido como uma 12-tupla $\langle S, \rho, \Psi, \delta, H, \tau, LSC, T_hist, V, C, E, T, Ac, Ts \rangle$, representando, respectivamente, estados, função de hierarquia, função de tipo de decomposição, função "default", conjunto de símbolos de história, função de história temporal, função de relógios locais, função de registradores de história temporal, conjunto de expressões, conjunto de condições, conjunto de eventos, conjunto de transições, conjunto de ações e conjunto de transições $M:N$ sincronizadas.

- **P: Info**³ denota o conjunto de páginas da forma $\langle c, t \rangle \mid c, t \in Info$ representando, respectivamente, o conteúdo e o título de uma página. Como definido no modelo HMBS, o conjunto **P** inclui a página nula, sem qualquer conteúdo ou título. Essa página pode ser utilizada com estados que não modelam a apresentação da informação, funcionando como artifícios de especificação, modelando a passagem do tempo, por exemplo.

- **Anc** \subset **Info** é o conjunto de âncoras do hiperdocumento. Uma âncora é uma unidade de informação que difere de uma página por, normalmente, conter uma porção reduzida de informação, como uma palavra, por exemplo.

- **M: S_s ↔ P** é uma relação que associa estados do subconjunto **S_s**, definido a seguir, a páginas do hiperdocumento.

$$S_s: \{x \in S \mid \psi(x) = OR \vee \rho(x) = \emptyset\}$$

A definição da relação **M** da forma **S_s ↔ P** é baseada na notação do método **Z** [Spi92]. A mesma definição pode ser escrita como **M: 2^{S_s} × P**

- **A: E × T × P → Anc** é a função que permite associar uma âncora a cada evento contido no rótulo de uma determinada transição de um *hyperchart*. Além disso, a âncora deve ser contextualizada por uma página. As restrições sobre a função **A** são definidas a seguir:

Seja $t \in T$ uma transição do *hyperchart* na forma $t = (X, r, Y)$, onde **X** é o conjunto de nós de origem, **Y** é o conjunto de nós de destino e r é o rótulo de t .

Seja $rot: T \times E \rightarrow \{true, false\}$ uma função que verifica se um evento primitivo e_n faz parte do rótulo de uma transição t_m , tal que $rot(t_m, e_n) = true$ se e_n faz parte do rótulo de t_m e $rot(t_m, e_n) = false$ se e_n não faz parte do rótulo de t_m .

então: $A(e, t, p) = anc \mid anc \in Anc \Rightarrow rot(t, e) = true \wedge \exists \langle s, p \rangle \in M \wedge (\exists subs \in X \mid s \in \rho(subs))$, ou seja, se um evento, uma transição e uma página são mapeadas para uma âncora, então é necessário que esse evento esteja presente no rótulo da transição e que exista um estado mapeado para uma página, de forma que esse estado pertença ao conjunto de origem da transição ou seja substado de um outro que pertence a esse conjunto.

- **L** é o nível de visibilidade do hiperdocumento, exatamente como definido pelo modelo HMBS.

- Pch é o conjunto de *Pchannels* (“Presentation Channels”), ou canais de apresentação, que são abstrações através das quais é possível especificar os requisitos relativos à apresentação da informação contida nas páginas.

- Cch é o conjunto de *Cchannels* (“Communication Channels”), ou canais de comunicação, que são abstrações da interface com o subsistema de comunicação subjacente, para o caso de se estar especificando uma aplicação distribuída.

- $V_p: P \times S_S \rightarrow Pch$ é uma função que associa pares formados pelas páginas do hiperdocumento e pelos estados a que estão associadas a *Pchannels* que especificam suas apresentações. A função V_p possui a seguinte restrição:

$$\forall p_1, p_2, \dots, p_n \in P, \forall s \in S_S, \forall pch_1, pch_2, \dots, pch_n \in Pch,$$

$$V_p(p_1, s) = pch_1 \wedge V_p(p_2, s) = pch_2 \wedge \dots \wedge V_p(p_n, s) = pch_n \Leftrightarrow pch_1 = pch_2 = \dots = pch_n,$$

ou seja, não é permitido que vários pares página-estado envolvendo o mesmo estado sejam associados, simultaneamente, a *Pchannels* diferentes.

- $V_a: Anc \times P \rightarrow Pch$ é uma função que associa um par formado por uma âncora e a página a que ela está associada a *Pchannels* que as apresentam e coletam os estímulos sobre elas. Essa função é restrita pelas seguintes relações:

$$\forall a \in Anc, \forall p \in P, \forall pch \in Pch$$

$$V_a(a, p) = pch \Leftrightarrow \exists e \in E \wedge \exists t \in T \mid A(e, t, p) = a,$$

ou seja, se uma âncora e uma página estão associadas a um *Pchannel*, então essa âncora deve estar associada a um evento, a uma transição e a essa página pela relação A .

$$\forall a_1, a_2, \dots, a_n \in Anc, \forall p_1, p_2, \dots, p_n \in P, \forall pch_1, pch_2, \dots, pch_n \in Pch$$

$$\forall t \in T, \forall e \in E,$$

$$(V_a(a_1, p_1) = pch_1 \mid A(e, t, p_1) = a_1) \wedge (V_a(a_2, p_2) = pch_2 \mid A(e, t, p_2) = a_2) \wedge \dots \wedge (V_a(a_n, p_n) = pch_n \mid A(e, t, p_n) = a_n) \Leftrightarrow pch_1 = pch_2 = \dots = pch_n,$$

ou seja, todas as âncoras relacionadas a páginas diferentes mas ao mesmo par evento-transição devem ser associadas ao mesmo *pchannel*.

- $Ch: Pch \rightarrow Cch \cup \emptyset$ é a função que associa um *Cchannel* a cada *Pchannel* utilizado pelo hiperdocumento. Se a aplicação é local, o *Pchannel* pode ser associado ao *cchannel* nulo, ou seja, $Ch(pch) = \emptyset$.

Essas são as definições de um hiperdocumento XHMBS, procurando atender aos requisitos ainda não satisfeitos pelo modelo HMBS. As considerações sobre as características do XHMBS são apresentadas na próxima seção.

5.3 As características do XHMBS

5.3.1 Melhor definição de páginas

A idéia do que é uma página em XHMBS permanece a mesma do modelo HMBS. No entanto, na definição formal do novo modelo, utilizou-se uma melhor caracterização desse elemento do hiperdocumento. Dessa forma, a partir de um conjunto básico *Info*, o qual representa as porções de informação em sua forma mais pura, define-se que uma página é um par composto de duas porções básicas de informação: um conteúdo e um título. Essa definição torna mais clara a noção de como uma página é estruturada, permitindo, por exemplo, que algoritmos de acesso direto às páginas sejam especificados sobre uma base formal bem estabelecida.

5.3.2 Várias páginas associadas ao mesmo estado

O mapeamento M , definido pelo HMBS como uma função, faz com que um estado seja associado a uma única página, o que constitui uma limitação ao modelo para o caso em que a aplicação deva possuir várias versões da mesma informação, as quais podem ser visualizadas em uma mesma seção de navegação.

O XHMBS permite que várias páginas sejam associadas a um mesmo estado, de forma que, se houver várias versões da mesma informação a serem disponibilizadas em uma mesma seção de navegação, a estrutura do hiperdocumento não precisará ser alterada, ficando por conta do sistema de apresentação, representado no modelo pelos *Pchannels*, a tarefa de apresentar a página correta.

Essa nova característica é obtida através da modificação do tipo do mapeamento M , que deixou de ser uma função ($M: S_s \rightarrow P$), para ser uma relação binária, ou o conjunto dos pares estado-página ($M: S_s \leftrightarrow P$). Essa nova possibilidade deu origem a uma modificação da relação de visibilidade V , a qual foi dividida em duas novas relações, apresentadas adiante.

5.3.3 Âncoras como unidades independentes da estrutura

A idéia original do HMBS de que uma transição entre estados representa uma ligação do hiperdocumento foi preservada. No entanto, no HMBS, por convenção, os eventos contidos nos rótulos associados às transições dos statecharts fornecem o rótulo da respectiva âncora no hiperdocumento. Essa abordagem, além de ser desprovida de um conceito formalizado na definição do modelo, faz com que o autor do hiperdocumento se preocupe com o conteúdo da informação enquanto define a estrutura da aplicação, o que é indesejável para o processo de autoria, visto que se tratam de tarefas distintas: especificação de ligações (estrutura) e definição de âncoras (conteúdo). Assim, com a separação desses dois processos de autoria, a especificação se torna menos acoplada, facilitando a implementação e a manutenção do hiperdocumento.

O XHMBS soluciona esse problema através da introdução de um conjunto de âncoras *Anc* e de um mapeamento A que as associa aos eventos definidos no hyperchart. Uma âncora, segundo o XHMBS, é definida, assim como as páginas, sobre o conceito de porções básicas de informação. Dessa forma, diz-se que algumas das porções básicas de informação do conjunto *Info* são âncoras do conjunto *Anc*. Em termos matemáticos, a relação de continência é a

representação mais apropriada para esse caso. O mapeamento A equivale, para âncoras, ao mapeamento M para páginas, ou seja, através da função A o conteúdo de uma âncora é associado ao seu correspondente em nível de estrutura.

Para garantir maior flexibilidade ao especificador e , ao mesmo tempo, garantir consistência com a utilização de versões diferentes da informação, a função A é definida sobre o domínio $E \times T \times P$, ou seja, uma âncora é associada a um evento contido no rótulo de uma determinada transição e , além disso, contextualizada para uma página específica. Essa definição permite que o mesmo evento seja utilizado para representar âncoras diferentes, dependendo do contexto em que se encontrar. Contexto, nesse caso, refere-se tanto à estrutura quanto ao conteúdo da informação, ou seja, à ligação determinada por uma transição e à página associada a um estado contido na hierarquia do conjunto de origem dessa transição. O uso de âncoras diferentes para versões diferentes do conteúdo da informação está associado à inclusão da página para se determinar uma âncora. Assim, se houver uma página escrita em inglês e outra em português, ambas associadas a um mesmo estado do *hyperchart*, serão definidas duas âncoras associadas a um mesmo evento contido no rótulo de uma mesma transição, mas, ainda assim, correspondentes a cada uma daquelas páginas.

Além da flexibilização do processo de autoria, sob pena de algum aumento da sua complexidade, o mapeamento entre eventos, transições e âncoras garante ao XHMBS a capacidade de satisfazer completamente o requisito de separação entre estrutura e conteúdo da informação, desvinculando o conteúdo de uma âncora de sua representação correspondente no *hyperchart*.

5.3.4 *Pchannels* e *Cchannels*

Os "readers" do HMBS foram substituídos por abstrações mais robustas, com maior capacidade de representação: os *Pchannels* e os *Cchannels*. Um *Pchannel*, ou canal de apresentação, é uma abstração abrangente de um dispositivo de apresentação que deve incluir atributos de apresentação de alto nível e mecanismos de coordenação espacial, entre outros. Se o hiperdocumento é distribuído e, portanto, é necessária a utilização de uma rede de computadores, a especificação dos requisitos de comunicação é realizada através de *Cchannels*, ou canais de comunicação, que abstraem a interface da aplicação com o sistema de comunicação subjacente.

A utilização de canais pelo XHMBS, apesar de inspirada no modelo Amsterdã [Har94], difere das propostas desse modelo por dividir os canais em grupos funcionais distintos, ou seja, o grupo de canais de apresentação e o grupo de canais de comunicação. Além disso, somente serão utilizados pares de canais de apresentação e de comunicação quando a apresentação for distribuída. Dessa forma, evita-se que objetos de especificação tenham características incompatíveis com o domínio da aplicação. Um exemplo desse tipo de inconveniência seria o uso de um só tipo de canal para especificar a apresentação e a obtenção de dados distribuídos. Nesse caso, se aplicação fosse local, todos os atributos do canal relativos à comunicação seriam desconsiderados, mas estariam, ainda assim, contidos na especificação.

Além da divisão dos canais em dois grupos funcionais, o XHMBS trata esses elementos de especificação segundo uma abordagem orientada a objetos, no sentido de que devam existir classes de canais *Pchannel* e *Cchannel*, as quais podem ser instanciadas conforme as necessidades do autor. Assim, um objeto *Pchannel* é uma instância de uma classe de canais

especializados em apresentar um certo tipo de informação. Dessa forma, não deverá existir um canal de apresentação de determinado tipo, mas tantos quantos forem necessários para a implementação da aplicação.

É interessante notar, ainda, que, devido à necessidade de receber eventos do ambiente, como quando utilizados por âncoras, a dimensão dos *Pchannels* passa a abranger todo o mecanismo de interação com o usuário, podendo funcionar como especificações funcionais das interfaces de entrada e saída da aplicação, o que não ocorre com o modelo HMBS, em que não há forma de se especificar, por exemplo, a apresentação de uma âncora. Em termos mais práticos, a relação estabelecida entre os estímulos recebidos por um *Pchannel* e os eventos definidos no *hyperchart* pode ser dada por alguns dos atributos desse *Pchannel*. Assim, as interações com o usuário podem ser melhor especificadas pelo XHMBS do que com o modelo HMBS.

No contexto deste trabalho, são definidas cinco classes de *Pchannels*: *VidPchannel*, para vídeo; *AudPchannel*, para áudio; *TxtPchannel*, para texto; *ImgPchannel*, para imagens e *AncPchannel*, para âncoras visuais (texto ou imagem). Certamente, no caso de se implementar um ambiente de desenvolvimento que suporte o XHMBS, podem ser criadas outras classes de *Pchannels* que suportem outros tipos de informação. Por exemplo: pode-se achar interessante dividir a classe *VidPchannel* em várias outras classes capazes de apresentar padrões mais específicos desse tipo de informação. Essa solução poderia tirar proveito da abordagem orientada a objetos e utilizar mecanismos de especialização / generalização, aproveitando-se, conseqüentemente, dos mecanismos de herança desse paradigma.

5.3.4.1 Características comuns aos *Pchannels*

O objetivo deste trabalho não é o de apresentar versões definitivas de cada *pchannel* ou *cchannel*, mas de introduzir esse conceito no âmbito do XHMBS. Possivelmente, a adição de conhecimentos adquiridos em trabalhos relativos à implementação de aplicações hipermedia distribuídas seja um passo na direção da definição classes mais apropriadas desses elementos de especificação. Por facilidade didática, a definição dos *pchannels* utilizados neste trabalho toma por base algumas características comuns a essas classes de objetos. A tabela 5.1 apresenta os atributos e métodos que devem estar presentes em qualquer *pchannel*.

Atributos	Descrição
<i>Tipo_mídia</i>	Tipo de mídia apresentado pelo <i>pchannel</i> (vídeo, áudio, imagem, texto ou âncora visual).
<i>Tipo_fonte (TR, PO)</i>	Especifica se os dados vêm de uma fonte de tempo real ou pré-orquestrada.
<i>Localização</i>	Define a localização do <i>pchannel</i> com relação ao dispositivo de apresentação. Para as mídias visuais, trata-se da definição de um retângulo ($\langle x_1, y_1 \rangle; \langle x_2, y_2 \rangle$). Para áudio, definem-se os autofalantes em que a informação será apresentada. A localização, juntamente com o tipo da mídia, define uma instância de um <i>pchannel</i> .

<i>Título (dir, esq, topo, base, invisível)</i>	Posição do título de uma página, caso se deseje que ele seja exibido. Este atributo não existe para âncoras visuais, já que estas não possuem títulos.
<i>Exclusividade (true, false)</i>	Define a exclusividade de um <i>pchannel</i> , isto é, se poderão ocorrer alocações simultâneas dos mesmos recursos utilizados por esse <i>pchannel</i> . Se a exclusividade for <i>true</i> , então, o atributo <i>prioridade</i> definirá a política de utilização dos recursos compartilhados pelos <i>pchannels</i> .
<i>Prioridade (normal, preferencial)</i>	Este atributo define a prioridade de apresentação dos dados de um conjunto de <i>pchannels</i> que alocam os mesmos recursos simultaneamente. Se um <i>pchannel</i> com prioridade <i>normal</i> está apresentando seus dados associados e um outro <i>pchannel</i> com prioridade <i>preferencial</i> tenta alocar os mesmos recursos, os dados do último serão apresentados sobre os do primeiro. Se os <i>pchannels</i> possuírem a mesma prioridade, pode-se optar por obedecer à ordem de solicitação de alocações ou gerar uma mensagem de erro para os dispositivos de apresentação.
<i>Origem (local, cchannel)</i>	Especifica o local de origem de uma página ou âncora. Se a origem for <i>local</i> , nenhuma especificação adicional é necessária. No entanto, se os dados vierem de algum local de origem distribuído pela rede, então o <i>cchannel</i> a que esse <i>pchannel</i> estiver associado será especificado.
<i>Velocidade (quadros / segundo)</i>	Velocidade de apresentação dos dados por um <i>pchannel</i> . A velocidade é dada em termos de quadros por segundo. No entanto, as interações de alteração de velocidade envolvem o relógio global do <i>hyperchart</i> e não os <i>pchannels</i> . A velocidade de apresentação, para os <i>pchannels</i> , representa o "throughput" real do sistema de apresentação, ou seja, a taxa com que os dados são apresentados pelo canal.
<i>Eventos</i>	São os eventos gerados pelo <i>pchannel</i> e que podem ser correspondentes aos eventos definidos no <i>hyperchart</i> . Os principais eventos definidos são <i>end</i> (fim da apresentação de uma página ou âncora) e <i>select</i> (seleção de uma âncora)
<i>Versões</i>	Versões do conteúdo da informação. Cada versão corresponde a um conjunto de páginas, as quais serão exibidas conforme a ativação dos estados do <i>hyperchart</i> . Uma das versões é definida como "default", ou seja, na primeira vez que a informação tiver que ser apresentada, se nenhuma das versões for selecionada, essa versão será apresentada.
<i>Dispositivos</i>	Descrição dos dispositivos de apresentação que o <i>pchannel</i> demanda.
Métodos	Descrição
<i>Allocate_ch ()</i>	Método que aloca um <i>pchannel</i> . Como esses métodos serão ativados por ações do <i>hyperchart</i> , o nome é dado em inglês, mantendo a compatibilidade com a linguagem.
<i>Release_ch ()</i>	Método que desaloca um <i>pchannel</i> .

<i>Iniciar ()</i>	Método que inicia a apresentação de uma página ou âncora que é submetida ao <i>pchannel</i> .
<i>Trocar_versão ()</i>	Método que possibilita ao usuário mudar a versão do hiperdocumento em tempo de navegação.
<i>Set_ch ()</i>	Método que permite alterar os valores de alguns atributos do <i>pchannel</i> .
<i>Liga ()</i>	Método que permite ligar (tornar visível) um canal que tiver sido desligado.
<i>Desliga ()</i>	Método que permite desligar (tornar invisível) um canal que estiver ligado.

Tabela 5.1 - Atributos e Métodos de um Pchannel genérico

5.3.4.2 Características particulares de cada pchannel

Alguns atributos e métodos de uma determinada classe de *pchannels* são estritamente dependentes do tipo de mídia que esse objeto pode apresentar. Assim, além das características comuns aos *pchannel*, apresentadas na tabela 5.1, cada um dos *pchannels*, em particular, pode possuir características adicionais ou, até mesmo, redefinir certas características, principalmente alguns métodos. Para os *pchannels* definidos neste trabalho, tem-se as caracterizações apresentadas a seguir:

VidPchannel

A classe *VidPchannel* possui alguns atributos específicos para a especificação de atributos globais de apresentação de cadeias de vídeo. Além disso, os métodos *allocate_ch* e *set_ch* são definidos de acordo com os atributos dessa classe. Dessa forma, a tabela 5.2 apresenta as características particulares da classe *VidPchannel*.

Atributos	Descrição
<i>Brilho (0..10)</i>	Define o brilho das imagens.
<i>Contraste (0..10)</i>	Define o contraste das imagens.
<i>Cor (0..10)</i>	Define a intensidade das cores utilizadas pelas imagens.
<i>Eventos (end)</i>	Define que o único evento gerado pelo <i>pchannel</i> é o de fim da apresentação dos dados.

Métodos	Descrição
<i>Allocate_ch</i> (<i>localização</i> [<x1,y1>, <x2,y2>],..., <i>brilho</i> , <i>cor</i> , <i>contraste</i>)	Sobrecarga do método <i>Allocate_ch</i> para trabalhar com vídeo.
<i>Set_ch</i> (<i>localização</i> ..., <i>brilho</i> , <i>cor</i> , <i>contraste</i>)	Sobrecarga do método <i>Set_ch</i> para trabalhar com vídeo. Deve-se fornecer o nome do atributo e seu novo valor.

Tabela 5.2 - Características particulares dos *VidPchannels*

AudPchannel

A classe *AudPchannel* é dedicada à especificação de atributos globais de apresentação de cadeias de áudio. Como ocorre com os *VidPchannels*, os métodos *allocate_ch* e *set_ch* são definidos de acordo com os atributos dessa classe de *pchannels*. A tabela 5.3 apresenta as características particulares da classe *AudPchannel*.

Atributos	Descrição
<i>Volume</i> (0..10)	Define o volume do som.
<i>Estéreo</i> (<i>true</i> , <i>false</i>)	Especifica se a reprodução do som será mono ou estereofônica.
<i>Balanco</i> (-5..+5)	Se o atributo <i>estéreo</i> tiver valor <i>true</i> , define a distribuição do som entre as duas vias de reprodução (direita e esquerda).
<i>Graves</i> (0..10)	Define a intensidade dos sons de frequência mais baixa.
<i>Agudos</i> (0..10)	Define a intensidade dos sons de frequência mais alta.
<i>Eventos</i> (<i>end</i>)	Define que o único evento gerado pelo <i>pchannel</i> é o de fim da apresentação dos dados.
Métodos	Descrição
<i>Allocate_ch</i> (<i>localização</i> [alto-falante,..., alto-falante_n],..., <i>volume</i> , <i>estéreo</i> , <i>balanco</i> , <i>graves</i> , <i>agudos</i>)	Sobrecarga do método <i>Allocate_ch</i> para trabalhar com áudio.
<i>Set_ch</i> (<i>localização</i> ..., <i>volume</i> , <i>estéreo</i> , <i>balanco</i> , <i>graves</i> , <i>agudos</i>)	Sobrecarga do método <i>Set_ch</i> para trabalhar com áudio. Deve-se fornecer o nome do atributo e seu novo valor.

Tabela 5.3 - Características particulares dos *AudPchannels*

TxtPchannel

Os *TxtPchannels* são dedicados à apresentação de dados textuais. Assim, para fins didáticos, somente alguns novos atributos foram adicionados ao conjunto inicial. As características particulares da classe *TxtPchannel* são apresentadas na tabela 5.4. Um detalhe

importante é que o atributo *velocidade* não é definido para as classes *TxtPchannel*, *ImgPchannel* e *AncPchannel*, já que essas mídias são discretas.

Atributos	Descrição
<i>Fonte_letra</i> (<i>Roman, Arial, Courier,...</i>)	Define um fonte de letras para o texto apresentado por este <i>pchannel</i> em particular.
<i>Tamanho_fonte</i> (8..20)	Define o tamanho do fonte de letra.
<i>Cores</i> (<i>fundo, texto</i>)	Define as cores de fundo e de letra para o texto.
<i>Scroll</i> (<i>[esq, dir],[topo, base]</i>)	Define a existência de barras de rolagem para o texto, tanto verticais (à esquerda ou à direita), quanto horizontais (acima ou abaixo do texto).
Métodos	Descrição
<i>Allocate_ch</i> (<i>localização[<x1,y1>, <x2,y2>]..., fonte_letra, tamanho_fonte, cores, scroll</i>)	Sobrecarga do método <i>Allocate_ch</i> para trabalhar com texto.
<i>Set_ch</i> (<i>localização, ..., fonte_letra, tamanho_fonte, cores scroll</i>)	Sobrecarga do método <i>Set_ch</i> para trabalhar com texto. Deve-se fornecer o nome do atributo e seu novo valor.

Tabela 5.4 - Características particulares dos *TxtPchannels*

ImgPchannel

Para permitir a especificação da apresentação de imagens, são definidos os atributos particulares da classe *ImgPchannel*. Para os exemplos de *pchannels* apresentados neste trabalho, tanto os *VidPchannels* quanto os *ImgPchannels* possuem os mesmos atributos particulares. A diferença entre eles é justamente a ausência do atributo *velocidade* nos *ImgPchannels*.

AncPchannel

A classe *AncPchannel* define as características de *pchannels* especializados em apresentar âncoras de natureza visual, mais especificamente, textos e imagens. Assim, pode-se dizer que as características particulares de um objeto *AncPchannel* são a união daquelas definidas para os *TxtPchannels* e para os *ImgPchannels*. Além disso, o atributo *eventos* recebe a definição do evento *select* que sinaliza um estímulo sobre o *pchannel*, sendo equivalente à geração do evento correspondente àquela âncora (definido pelos mapeamentos V_4 e A). Para permitir que o evento *select* seja gerado pelo *pchannel*, define-se também o método *Select ()*, que simplesmente implementa a recepção do estímulo dado pelo usuário.

5.3.4.3 Cchannels

Os *Cchannels* são objetos instanciados a partir de classes que definem as interfaces de um sistema de comunicação genérico. Essas abstrações devem incluir atributos e métodos que

permitam ao desenvolvedor especificar operações de comunicação, principalmente aquelas relacionadas com a negociação da qualidade de serviço das conexões solicitadas. Neste trabalho é definida uma única classe de *Cchannels*, denominada *GenCchannel* ("General" *Cchannel*), a qual permite a especificação dos principais atributos de qualidade de serviço de uma conexão no nível de transporte em uma rede genérica, segundo o modelo OSI/ISO de referência. A tabela 5.5 apresenta as características da classe *GenCchannel*.

Atributos	Descrição
<i>Origem</i>	Endereços dos locais físicos que compõe um local de origem tratado pelo <i>CChannel</i> .
<i>Atraso (ms)</i>	Especifica qual é o retardo de trânsito permitido, ou seja, qual é o tempo máximo tolerável para que uma mensagem chegue do local de origem ao local de destino.
<i>Throughput (bytes/s)</i>	Especifica o "throughput" da rede em termos de quantos bytes devem ser transportados em um segundo de transferência de dados.
<i>Erros_bit (bits/s)</i>	Especifica quantos erros de bits são toleráveis por segundo de transferência.
<i>Erros_pacote (pacotes/s)</i>	Especifica quantos erros de pacotes podem ocorrer em um segundo de transferência.
Métodos	Descrição
<i>Conectar (origem, atraso, throughput, erros_bit, erros_pacote)</i>	Método que permite o estabelecimento da conexão e a comunicação de quais são os parâmetros de qualidade de serviço desejados.
<i>Desconectar ()</i>	Método que modela a desconexão de um <i>cchannel</i> .

Tabela 5.6 - Características de um *GenCchannel*

O mapeamento *Ch* permite que um *pchannel* seja associado a um *cchannel*. Assim, quando a apresentação for distribuída, os requisitos de apresentação dos *pchannels* serão associados aos requisitos de comunicação dos *cchannels*, completando a especificação. Em termos da semântica de navegação, isso significa que quando um *pchannel* for requisitado para a apresentação de alguma informação cujo local de origem estiver distribuído pela rede, os dados chegarão ao *pchannel* através do *cchannel* correspondente.

5.3.5 As relações de visibilidade

O mapeamento *V* do modelo HMBS foi dividido em dois mapeamentos: V_s e V_r . O mapeamento V_r é equivalente à relação de visualização do modelo HMBS, ou seja, define as associações entre páginas e estados. No entanto, apesar da equivalência, em termos funcionais entre os dois mapeamentos, o mapeamento V_r possui restrições que garantem a correção da utilização da nova definição de *M*. Essas restrições, como apresentado na definição formal do modelo, dizem que páginas diferentes associadas a um mesmo estado nunca poderão ser

associadas a *pchannels* diferentes. Isso se deve ao fato de que o *pchannel* é o elemento responsável pela seleção de uma das páginas associadas a um mesmo estado. Se duas ou mais páginas associadas ao mesmo estado pudessem ser relacionadas com *pchannels* diferentes, não seria possível determinar qual dessas páginas seria apresentada quando o estado correspondente no *hyperchart* estivesse ativo. Mais do que isso, o uso de *pchannels* diferentes permitiria que duas ou mais páginas, cada uma associada a um *pchannel*, fossem apresentadas simultaneamente, o que incorporaria ao modelo uma outra forma de se especificar a apresentação concorrente da informação, podendo contradizer a especificação estrutural do *hyperchart*. A figura 5.1 apresenta um esquema de mapeamentos M e V_F em que três páginas são associadas a um mesmo estado. Deve-se notar nesse exemplo que o mapeamento V_F faz com que as três páginas sejam associadas ao mesmo *pchannel*.

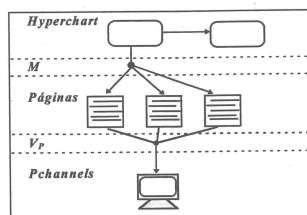


Figura 5.1 - Mapeamentos M e V_F combinados

O mapeamento V_a , que permite que uma âncora seja associada a um *Pchannel*, é a forma com que o XHMBS provê independência de dispositivo para as âncoras, assim como ocorre com as páginas através do mapeamento V_F . Isso quer dizer que, através do uso de *Pchannels* específicos, uma âncora pode ser apresentada tanto por uma palavra grifada exibida junto ao conteúdo de uma página como por um dispositivo sensível ao tato ou um microfone, por exemplo. Isso faz com que aumente a possibilidade de aplicações dos hiperdocumentos, uma vez que, com a utilização de objetos de diferentes mídias como receptores de estímulos externos, os recursos de interação com o usuário se tornam muito mais sofisticados e, dessa forma, atraentes.

As restrições sintáticas sobre o mapeamento V_a exigem que, se uma âncora e uma página forem associadas a um *pchannel*, essa âncora deverá estar associada a um evento e a uma transição contextualizados por aquela página e, além disso, todas as âncoras associadas a páginas diferentes e ao mesmo evento contido no rótulo de uma mesma transição deverão estar associadas ao mesmo *pchannel*. Assim, a figura 5.2 mostra um exemplo com o uso de âncoras.

Note-se que nesse exemplo há duas versões da mesma informação, demandando, portanto, duas páginas, as quais estão associadas a um mesmo *pchannel* através do mapeamento V_F . O estado ao qual as duas páginas estão associadas é origem de uma transição associada a um evento e . Essa transição e esse evento são mapeados, juntamente com cada página, para suas respectivas âncoras. Assim, para cada versão da informação determinada por uma página, há uma versão de âncora correspondente. Da mesma forma que ocorre com as

páginas, as duas âncoras são mapeadas para um mesmo *pchannel*, garantindo que as apresentações serão consistentes com as versões especificadas.

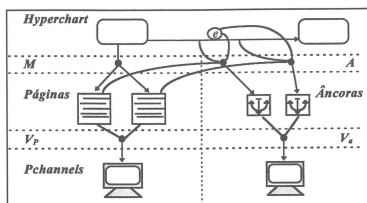


Figura 5.2 - Mapeamentos com âncoras e páginas

5.3.6 A semântica de navegação do XHMBS

Uma vez apresentadas as novas características do modelo, resta definir a sua semântica de navegação. Assim, nesta subsecção são apresentadas as extensões à semântica de navegação do modelo HMBS que suportam as definições apresentadas pelo XHMBS.

A diferença inicial entre as semânticas de navegação dos dois modelos é o uso de statecharts pelo HMBS e de *hypercharts* pelo XHMBS. A idéia de que a informação é apresentada conforme as configurações assumidas pelo modelo subjacente (neste caso, um *hyperchart*) permanece inalterada. O que muda é a forma com que a informação associada ao *hyperchart* é apresentada e isso se deve às novas definições de elementos e de mapeamentos do modelo XHMBS. Dessa forma, a semântica de navegação aqui apresentada é baseada no comportamento da aplicação quando uma configuração de estados do *hyperchart* é fornecida.

Dado o conjunto SC de estados que determina uma configuração do *hyperchart* subjacente, tem-se que a informação será apresentada segundo os seguintes passos:

$\forall s \in SC, \exists Cp = \{p_i \in P \mid M(s) = p_i \wedge p_i \neq NULL\}$, onde NULL denota a página nula $p_{null} = \langle \emptyset, \emptyset \rangle$.

- Seja $pch_i \in Pch$ o *pchannel* que apresenta as páginas contidas em um conjunto Cp tal que $\forall p_i \in Cp, Vp(p_i, s) = pch_i$.
- Executa-se: $pch_i.Iniciar()$, para que a apresentação ocorra.

A versão da informação a ser visualizada é determinada pelo usuário e pode ser modificada em qualquer momento através de uma mensagem $pch_i.Trocar_versão(v)$. Quando uma versão é selecionada em um *pchannel*, o sistema de apresentação cuida para que todos os outros *pchannels* apresentem a mesma versão, difundindo mensagens $Trocar_versão(v)$ para todos os *pchannels*. É de responsabilidade do especificador determinar a correspondência correta entre as versões especificadas em cada *pchannel*.

Quando a configuração do *hyperchart* for a inicial, a versão "default" de cada *pchannel* será a utilizada.

- Se $pch_i.Source \neq local$, então se deve executar $cchannel_i.Conectar(q,s)$, onde $cchannel_i = Ch(pch_i)$ e q,s é a lista de parâmetros de qualidade de serviço requeridos para a conexão.

A execução do método *Iniciar()* pelo *pchannel* é responsável pelo recebimento e apresentação dos dados, tenham eles vindo de uma fonte local ou remota.

- $\forall t = (Source, l, Target) \in T$ relevante à configuração de estados do *hyperchart*, tal que $s \in Source$ e l é da forma "ev / a", tem-se que

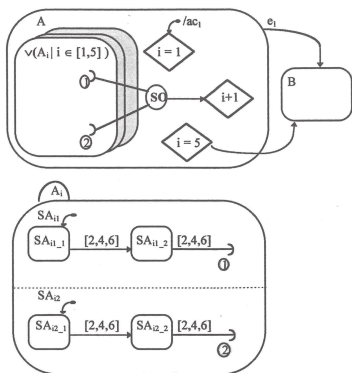
Para todo evento primitivo $e \in E$ contido na expressão *ev* com $p_i \in P$, $anc \in Anc$, $pch_k \in Pch \mid A(e, t, p_i) = anc \wedge V_s(anc, p_i) = pch_k$, se pch_k ainda não tiver sido alocado, executa-se $pch_k.Allocate_ch()$ e, de qualquer forma, $pch_k.Iniciar()$, sendo que todas as observações referentes às versões e uso de *cchannels* citadas acima são igualmente válidas para este caso.

Os *ancpchannels* (canais de apresentação especializados em âncoras) que se tornarem ativos terão habilitados seus métodos *Select()*, os quais coletarão os estímulos e os transmitirão à máquina de *hypercharts* subjacente, conforme a correspondência com os eventos aos quais as âncoras estiverem associadas. O sistema de apresentação deve disponibilizar uma forma de gerar esses eventos, de forma que uma nova configuração do *hyperchart* seja gerada, desencadeando novamente os procedimentos de apresentação descritos.

As definições relativas à semântica de navegação com níveis de visibilidade diferentes permanecem as mesmas encontradas no modelo HMBS. A exploração da hierarquia dos *hypercharts* através dos níveis de visibilidade e do mecanismo de visões hierárquicas é perfeitamente compatível com a nova semântica de navegação.

5.4 A notação do modelo XHMBS

Em termos de notação, o XHMBS conta com os recursos gráficos dos *hypercharts*, da mesma forma como ocorre com os *statecharts* no HMBS, além de tabelas que representam os mapeamentos definidos para o hiperdocumento (M, A, V_f, V_s, Ch). Assim, considerando um pequeno exemplo de *hyperchart*, como o da figura 5.3, pode-se ter a seguinte especificação para um hiperdocumento com duas versões.



$ac_1 = \text{"(img1.Allocate_ch(localização = } x_1, y_1, x_2, y_2, \text{ tipo_fonte = PO, título = topo, exclusividade = true, prioridade = preferencial, origem = cch, brilho = 6, cor = 5, contraste = 10, eventos = end, versões = } v_1 \rightarrow \{p_{i,v_1} | i \in N, 1 \leq i \leq 10\}, v_2 \rightarrow \{p_{i,v_2} | i \in N, 1 \leq i \leq 10\})\text{"}$

$(\text{aud1.Allocate_ch(localização = } a_{f_1}, a_{f_2}, \text{ tipo_fonte = PO, exclusividade = true, prioridade = preferencial, origem = cch}_2, \text{ volume = 8, estéreo = true, balanço = 0, graves = 10, agudos = 9, eventos = end, version = } v_1 \rightarrow \{p_{i,v_1} | i \in N, 11 \leq i \leq 20\}, v_2 \rightarrow \{p_{i,v_2} | i \in N, 11 \leq i \leq 20\})\text{"}$

Figura 5.3 - Exemplo simples de especificação em XHMBS

Considerando um conjunto de páginas $P = \{p_{i,v_1} | i \in N, 1 \leq i \leq 20\} \cup \{p_{i,v_2} | i \in N, 1 \leq i \leq 20\} \cup \{p_null\}$, um conjunto de âncoras $Anc = \{anc_{v_1}, anc_{v_2}\}$, três classes de *pchannels*: *ImgPchannel*, *AudPchannel* e *AncPchannel* e uma classe de *cchannel*: *GenCchannel*, pode-se ter os seguintes mapeamentos:

Mapeamento M	
A	p_null
B	p_null
SA _{i1}	p_null
SA _{i2}	p_null

Mapeamento M	
$SA_{i1_1} i \in N, 1 \leq i \leq 5$	$p_{1,v1} j \in N, 1 \leq j \leq 5$
$SA_{i1_1} i \in N, 1 \leq i \leq 5$	$p_{1,v2} j \in N, 1 \leq j \leq 5$
$SA_{i1_2} i \in N, 1 \leq i \leq 5$	$p_{1,v1} j \in N, 6 \leq j \leq 10$
$SA_{i1_2} i \in N, 1 \leq i \leq 5$	$p_{1,v2} j \in N, 6 \leq j \leq 10$
$SA_{i2_1} i \in N, 1 \leq i \leq 5$	$p_{1,v1} j \in N, 11 \leq j \leq 15$
$SA_{i2_1} i \in N, 1 \leq i \leq 5$	$p_{1,v2} j \in N, 11 \leq j \leq 15$
$SA_{i2_2} i \in N, 1 \leq i \leq 5$	$p_{1,v1} j \in N, 16 \leq j \leq 20$
$SA_{i2_2} i \in N, 1 \leq i \leq 5$	$p_{1,v2} j \in N, 16 \leq j \leq 20$

O mapeamento M é representado, nesse exemplo, com uma notação parametrizada que segue, basicamente, a mesma linha da parametrização de estados. Isso porque a utilização de estados parametrizados no *hyperchart* não faria sentido se não pudesse ser acompanhada do mesmo recurso em outras representações do modelo. Assim, através de índices, é possível especificar a parametrização dos mapeamentos de forma similar àquela utilizada em *hypercharts*.

Mapeamento A	
$e_1, t, p_{1,v1} t = (\{A\}, \{e_1\}, \{B\})$	anc_{v1}
$e_1, t, p_{1,v2} t = (\{A\}, \{e_1\}, \{B\})$	anc_{v2}

O mapeamento A utiliza páginas associadas ao subestado SA_{i1_1} , contido na hierarquia do conjunto de estados de origem de t ($\{A\}$). Assim, se a versão que estiver sendo utilizada for aquela correspondente à página $p_{1,v1}$, a âncora anc_{v1} será exibida e será o ponto de interface para a geração do evento e_1 . Por outro lado, se a versão em uso for aquela correspondente à página $p_{1,v2}$, então a âncora utilizada será anc_{v2} .

Mapeamento V_P	
$p_{1,v1} j \in N, 1 \leq j \leq 10$ associadas a estados do tipo SA_{i1_1} e SA_{i1_2}	$img1 : \text{ImgPchannel}$

$p_{1,v2} \mid j \in N, 11 \leq j \leq 20$ associados a estados do tipo $SA_{A1,1}$ e $SA_{A1,2}$ (versão 2)	img1 : <i>ImgPchannel</i>
$p_{1,v1} \mid j \in N, 1 \leq j \leq 10$ associados a estados do tipo $SA_{A2,1}$ e $SA_{A2,2}$	aud1 : <i>AudPchannel</i>
$p_{1,v2} \mid j \in N, 11 \leq j \leq 20$ associados a estados do tipo $SA_{A2,1}$ e $SA_{A2,2}$ (versão 2)	aud1 : <i>AudPchannel</i>

Para esse caso, foram utilizados somente dois canais de apresentação: um do tipo *ImgPchannel* (img1), e outro do tipo *AudPchannel* (aud1). O mapeamento V_p , como definido para esse exemplo, determina que as páginas associadas aos subestados de estados do tipo SA_{A1} devem ser mapeadas para o canal img1. Dessa forma, espera-se que a informação contida nessas páginas esteja na forma de imagens. Note-se que são definidos mapeamentos que levam páginas de ambas as versões ao mesmo canal. Analogamente, as páginas associadas aos subestados de estados do tipo SA_{A2} são mapeadas para o canal aud1, de forma que se espera que as informações contidas naquelas páginas estejam em forma de áudio.

Mapeamento V_a	
$anc_{v1}, p_{1,v1}$	<i>ancpch1</i> : <i>AncPchannel</i>
$anc_{v2}, p_{1,v2}$	<i>ancpch1</i> : <i>AncPchannel</i>

O mapeamento V_a , como definido acima, faz com que as duas âncoras, cada uma associada a uma página que a contextualiza, sejam mapeadas para um mesmo canal (*ancpch1*), de forma que a versão correta possa ser exibida em tempo de navegação.

Mapeamento Ch	
img1	<i>cch1</i> : <i>GenCchannel</i>
aud1	<i>cch2</i> : <i>GenCchannel</i>
<i>ancpch1</i>	<i>cch3</i> : <i>GenCchannel</i>

O mapeamento Ch define quais são os *channels* necessários para que as páginas e âncoras sejam recebidas pelos respectivos *pchannels* através da rede. Dessa forma, o hiperdocumento é completamente especificado. Como se pode notar na figura 5.3, as expressões de ações que tratam da alocação de canais são bastante longas, uma vez que

determinam todas as características de um ou mais *pchannels*. Além disso, como é possível notar, não há qualquer especificação referente à alocação de *Ancpchannels*, uma vez que esse procedimento está incorporado à semântica de navegação. Para evitar o uso de sentenças muito longas como as que são utilizadas na figura 5.3, pode-se optar por definir em tabelas nas quais são especificadas as instâncias de cada *pchannel* utilizado e simplesmente especificar o método na expressão de ações do *hyperchart*. Se o problema demandar a modificação de algum dos atributos via especificação, então somente as expressões de ações contendo mensagens *Set_ch()* serão utilizadas, aliviando graficamente a especificação. Por exemplo, a especificação do canal *ancpch₁* pode ser realizada através de uma tabela como a apresentada a seguir:

<i>ancpch₁</i>			
<i>fonte_letra</i>	<i>arial</i>	<i>contraste</i>	10
<i>tamanho_fonte</i>	10	<i>cores</i>	7
<i>scroll</i>	-	<i>eventos</i>	<i>end, select</i>
<i>brilho</i>	6	<i>versões</i>	$v_1 \rightarrow p_{1,v1} v_2 \rightarrow p_{1,v2}$

Para grandes aplicações é aconselhável o uso dessa abordagem de especificação de canais para que a documentação do projeto se torne mais legível. Finalmente, tabelas similares devem ser construídas para especificar os atributos dos *cchannels*, definindo os requisitos de comunicação desejados. Como exemplo, o *cchannel cch₁* poderia ser especificado segundo a tabela a seguir:

<i>cch₁</i>	
<i>origem</i>	143.206.12 (endereço IP de um servidor)
<i>atraso</i>	10 ms
<i>throughput</i>	10 Mbits/s
<i>erros_bit</i>	3 / s
<i>erros_pacote</i>	0,5 / s

5.5 Interações com o usuário no modelo XHMBS

Nesta seção são discutidas as interações com o usuário que podem ser especificadas através do modelo XHMBS. Essas interações podem ser disponibilizadas de duas formas:

através das ligações do hiperdocumento, permitindo a navegação, e através das interfaces multimídia oferecidas pelo sistema de apresentação através dos *pchannels*.

A modelagem das interações com o usuário pode ser realizada livremente com a notação de *hypercharts*, acrescentando-se uma ação que manipula o relógio global do *hyperchart* subjacente ao hiperdocumento, definida a seguir:

Seja H o conjunto dos *hypercharts*, N o conjunto dos números naturais e TU o conjunto das unidades de tempo.

$G_clock: H \rightarrow N \times TU \mid G_clock(h) = n / tu$ significa que ocorrem n passos a cada tu no *hyperchart* h . Assim, por exemplo, $G_clock(h) = 10 / s$ significa que a velocidade de execução do *hyperchart* h é de dez passos por segundo.

Se, inicialmente, $G_clock(h) = n / tu$, após a execução da ação especial $New_G_clock(ngc)$, $G_clock(h) = ngc / tu$. Ou seja, New_G_clock é uma ação que permite alterar a velocidade do relógio global de um *hyperchart*.

Como facilidade de especificação, presume-se que o sistema de apresentação possua alguns valores de velocidades do relógio global do *hyperchart* pré-definidos, a saber: "standard", que é a velocidade "default" do *hyperchart*; "skip_speed", que é a velocidade mais alta disponível, utilizada para a definição da interação "saltar", e "fast" que é a velocidade utilizada para apresentações rápidas. Além disso, podem ser oferecidos alguns fatores de alteração de velocidade, como "double", "half", os quais permitem, respectivamente, que a velocidade de execução do *hyperchart* seja duplicada ou reduzida pela metade.

De posse desse conjunto de definições, sugere-se o uso de alguns blocos de construção que facilitam o trabalho de especificação de interações com o usuário. Esses blocos de construção não são, de fato, parte do modelo XHMBS. Por outro lado, aconselha-se que esses padrões de especificação sejam utilizados sempre que possível, já que contêm muitas soluções relativas à especificação de interações com o usuário. As interações suportadas por esses blocos de construção são as mesmas propostas pelos blocos de construção do modelo TSPN_{UI} [Coo95], ou seja: "iniciar", "saltar", "inverter a direção de apresentação", "congelar", "descongelar", "alterar a velocidade de apresentação", "apresentação rápida", "terminar", "reiniciar (conforme a direção de apresentação)". Além dessas interações, chamadas primitivas, através de interfaces do sistema de apresentação, pode-se disponibilizar as interações compostas "saltar para trás", composta de "inverter a direção de apresentação" seguida de "saltar", e a interação de "apresentação rápida para trás", composta de "inverter a direção de apresentação" e "apresentação rápida".

As figuras 5.4 e 5.5 apresentam os blocos de construção sugeridos para o uso com o modelo XHMBS. Na figura 5.4, é apresentado um bloco de construção que pode ser utilizado em aplicações que demandam duas direções de apresentação com possibilidade de uso da interação "inverter a direção da apresentação", representada pelo evento *inv*. A ação utilizada juntamente com as transições associadas ao evento *inv* é uma modificação no valor do relógio local do estado de destino, de forma a fazer com que a permanência nesse estado seja igual ao tempo remanescente de apresentação no sentido contrário. A variável d é utilizada para modelar a direção de apresentação e, portanto, assume somente dois valores "0" (para frente)

e “1” (para trás). Assim, as transições associadas ao evento *inv* possuem uma ação associada que altera o valor de *d* de acordo com a direção assumida pela apresentação.

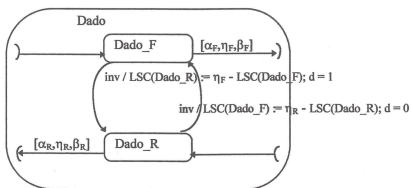


Figura 5.4 - Bloco de construção para dados apresentados em duas direções com interação de inversão

Na figura 5.5, apresenta-se um bloco de construção que corresponde aos níveis mais altos de abstração da especificação. Dessa forma, toda a apresentação deve ser encapsulada pelo estado *Apresentação*, de forma que as interações fiquem disponíveis em escopo global. Esse bloco de construção, com exceção das transições que manipulam o relógio global do *hyperchart* pode ser aplicado recursivamente em regiões específicas do hiperdocumento, de forma que suas apresentações possam ser controladas independentemente.

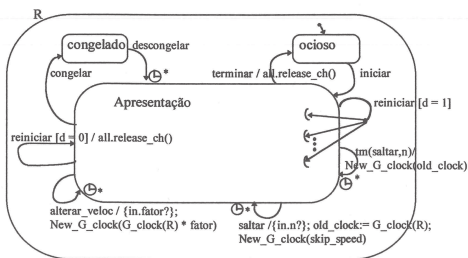


Figura 5.5 - Bloco de construção de mais alto nível

O par de interações “congelar” e “descongelar” é modelado por duas transições, cada uma associada a um evento. O estado *congelado* modela a situação da aplicação após a ocorrência do evento *congelar* a partir do estado *Apresentação*. A volta à apresentação, modelada pelo evento *descongelar* associado a uma transição com história temporal, faz com que a apresentação seja retomada a partir do mesmo ponto em que se encontrava quando foi

congelada. O fato de a aplicação estar no estado *congelado* não implica no fato de que a informação deixará de ser apresentada. Isso é garantido pela semântica de navegação do modelo, que só troca a informação apresentada pelos *pchannels* quando os mesmos são liberados ou quando um estado associado a uma página diferente da nula torna-se ativo, ativando o *pchannel* correspondente.

As interações “iniciar” e “terminar” são modeladas de forma similar às interações “congelar” e “descongelar”, no sentido de que há um estado que modela a situação da aplicação quando não há qualquer apresentação de dados e duas transições, cada uma modelando uma das interações. No entanto, não há qualquer entrada por história temporal, visto que a interação “iniciar” realmente inicia a apresentação como se ela nunca tivesse ocorrido antes, desconsiderando outras seções de navegação sobre aqueles dados. A transição associada ao evento *terminar*, que modela a interação “terminar”, tem a ação *all.release_ch*, que corresponde, por convenção, a uma difusão de mensagens *release_ch* para todos os *pchannels* ativos no estado *Apresentação*.

A interação “reiniciar”, como proposta nesse bloco de construção, é considerada para as duas direções de apresentação, modeladas pelo valor da variável *d*. Assim se a direção for para frente, a transição associada ao evento *reiniciar* e à condição “*d = 0*” irá disparar. Se a direção for para trás, então a transição que disparará será aquela associada ao evento *reiniciar* e à condição “*d = 1*”. A ação *all.release_ch* é utilizada com a transição associada à condição “*d = 0*” para garantir que não ocorram erros durante as alocações de canais especificadas dentro do estado *Apresentação*, devido ao reinício da apresentação. Para *d = 1* isso não é necessário, uma vez que os canais já deverão estar alocados.

A interação “alterar a velocidade da apresentação” é modelada pela transição associada ao evento *alterar_veloc*, cuja expressão de ações especifica que um dado externo é recebido (“{in.fator?”}), representando o fator de alteração da velocidade de apresentação *e*, além disso, que a velocidade do relógio global da aplicação é atualizada, o que é obtido com a expressão “New_G_clock(G_clock(Hyp) * fator)”, partindo do pressuposto de que Hyp seja o identificador do *hyperchart*. A expressão “{in.x?”} denota que o valor de uma certa variável *x* é lido do ambiente através do canal de entrada (*in*) do *hyperchart*.

Finalmente, a interação “saltar”, que corresponde a um salto de *n* passos na direção corrente, é modelada pelas transições associadas aos eventos *saltar* e *tm(saltar,n)*. O funcionamento dessa combinação de eventos é o seguinte: a ação associada ao evento *saltar* recebe o número de passos do salto (“{in.n?”}) e guarda o valor atual da velocidade do relógio global do *hyperchart* em uma variável (*old_clock*) e, então, faz com que o relógio global assumira a velocidade *skip_speed*, especialmente definida para essa interação. Presume-se que essa velocidade seja tão grande que, em termos de interface com o usuário, seja impossível a geração de qualquer evento externo. Dessa forma, a apresentação irá ocorrer a uma velocidade muito alta durante *n* passos (caracterizando o salto), após os quais a transição associada ao evento *tm(saltar,n)* disparará, de forma que a ação associada a ela restaurará o valor anterior da velocidade do relógio global, salvo em *old_clock*. É importante lembrar que o escopo das ações que modificam o relógio global é o mesmo em qualquer nível de abstração da especificação. Assim, é impossível acelerar somente uma parte da apresentação de um hiperdocumento. Isso quer dizer que as interações “alterar a velocidade da apresentação” e

“saltar” só devem ser especificadas no uso de mais alto nível do bloco de construção apresentado na figura 5.5. Uma possível solução para esse problema é a definição de relógios com escopo definido, podendo ser aplicados a certas hierarquias de estados e não a todo o *hyperchart*.

5.6 Um exemplo de especificação com XHMBS

Em [Coo95] é apresentado um exemplo de documento multimídia, o qual foi especificado segundo os modelos OCPN, XOCPN, DTPN, TSPN e TSPN_{UI}, todos baseados em redes de Petri e apresentados no capítulo 3 deste trabalho. As especificações através desses cinco modelos permitiram compará-los entre si em termos de complexidade, medida em função do número de símbolos utilizados.

Nesta seção, para ilustrar a utilização do modelo e compará-lo a outras propostas, é apresentada a especificação de um hiperdocumento segundo o XHMBS. Uma das partes desse hiperdocumento é a apresentação multimídia especificada em [Coo95]. Assim, para propósitos da comparação entre os modelos, somente a porção multimídia do hiperdocumento é considerada. A seção 5.6.1 apresenta a descrição textual do hiperdocumento e as seguintes a sua especificação em XHMBS.

5.6.1 Descrição do hiperdocumento

Há cinco grandes porções no hiperdocumento. Quando o usuário inicia a navegação lhe são apresentados dois textos, um após o outro, ao mesmo tempo que uma trilha sonora. Cada texto tem duração de cinco segundos, determinada por uma ligação temporal entre eles, ou seja, o usuário poderá seguir a ligação antes dos cinco segundos de apresentação do primeiro texto. A trilha sonora é representada por quatro quadros de áudio, os quais possuem dois segundos e trinta segundos cada um. O segundo texto e o quarto quadro de áudio devem terminar ao mesmo tempo, sendo seguidos de uma página de texto (um menu) com ligações para a introdução e para as outras três partes do hiperdocumento: uma galeria de fotos com cinco imagens sendo apresentadas simultaneamente até que o usuário siga a ligação que o leva de volta ao menu; uma página de texto com informações sobre o autor do hiperdocumento, a partir da qual, como ocorre com a galeria de fotos, a única ligação a seguir leva de volta ao menu e, finalmente, a apresentação multimídia, descrita na seção 5.6.2. Estando na página de introdução do hiperdocumento, o usuário pode interromper a apresentação e seguir direto para o menu através de uma ligação temporal. A ligação que leva do menu à apresentação multimídia também é temporal, de forma que, se após 10 segundos visitando o menu, o usuário não seguir nenhuma ligação, a apresentação multimídia começará a ser apresentada.

5.6.2 A apresentação multimídia

A apresentação é composta de seis seqüências de vídeo (SV), três seqüências de áudio (SA), duas imagens (I) e duas cadeias de texto (T). A apresentação tem duração total de trinta segundos. Os demais requisitos dessa apresentação são os seguintes:

1. Cada SV tem duração de cinco segundos;
2. Cada SA tem duração de dez segundos;

3. Cada I tem duração de dez segundos;
4. O "jitter" máximo permitido para cada imagem é de dois segundos;
5. O primeiro texto tem duração de dez segundos;
6. O segundo texto tem duração de vinte segundos
7. O "jitter" máximo permitido para cada cadeia de texto é de dois segundos;
8. A primeira SV, a primeira SA, a primeira I e o primeiro T começam a ser apresentados ao mesmo tempo;
9. Quando a primeira SV termina, a segunda começa;
10. A segunda SV e a primeira SA acabam ao mesmo tempo. Quando essas cadeias terminam, a terceira SV e a segunda SA começam a ser apresentadas ao mesmo tempo.
11. Quando a terceira SV termina, a quarta começa.
12. A quarta SV e a segunda SA acabam ao mesmo tempo. Quando elas terminam, a quinta SV e a terceira SA começam ao mesmo tempo.
13. Quando a quinta SA termina, a sexta começa a ser apresentada.
14. As SA e SV precisam ser sincronizadas a cada terço de segundo. O "jitter" máximo permitido é de setenta e cinco milissegundos para cada ponto de sincronização.
15. A primeira I e o primeiro texto terminam ao mesmo tempo. Quando eles terminam a segunda I e o segundo T começam a ser apresentados ao mesmo tempo.
16. A segunda imagem termina na metade do tempo de apresentação do segundo texto.
17. Quando a segunda I termina, a área de "display" para a imagem é liberada e fica assim até o final da apresentação.
18. A sexta SV, a terceira SA e o segundo T terminam ao mesmo tempo. A sexta SV e o segundo T permanecem na região de "display" até que o usuário solicite o término da apresentação ou siga a ligação que leva de volta ao menu.
19. Cada uma das cadeias de dados tem origem em um único local.
20. Todas as fontes de dados são pré-orquestradas.
21. O usuário pode interagir com a apresentação das seguintes formas:
 - **Iniciar:** O usuário requisita o início da apresentação. Essa interação com o usuário não é permitida depois de a apresentação começar.

- **Saltar para frente:** O usuário requisita um salto de n unidades de tempo na direção corrente de apresentação.
- **Saltar para trás:** O usuário requisita um salto de n unidades de tempo na direção de apresentação oposta à corrente.

As interações relativas a saltos não são permitidas quando a apresentação tiver terminado ou quando estiver congelada.

- **Reiniciar:** O usuário requisita o reinício da apresentação, a qual deve ser reiniciada de acordo com a direção corrente de apresentação. Essa interação não é permitida antes do início da apresentação.
- **Terminar:** O usuário requisita o término da apresentação. Após o término, o usuário só poderá requisitar a interação "iniciar". A interação "terminar" não é permitida antes da apresentação ter iniciado.
- **Congelar:** O usuário requisita que a apresentação seja congelada, ou seja, que uma pausa seja feita.
- **Descongelar:** O usuário requisita que a apresentação, que deverá estar congelada, seja retomada a partir do ponto em que estava quando houve a pausa.
- **Inverter a direção da apresentação:** O usuário requisita que a apresentação passe a ser feita na direção oposta à corrente. Essa interação não pode ocorrer sem que a apresentação tenha iniciado ou quando estiver congelada.
- **Alterar a velocidade da apresentação:** O usuário requisita uma modificação na velocidade de apresentação. As novas velocidades poderão ser tanto maiores quanto menores do que a velocidade corrente. Os fatores de escalamento são de $\frac{1}{2}$, 1 e 2, tanto para aumentar quanto para diminuir a velocidade. Essa interação não poderá ocorrer quando a apresentação estiver congelada.

Quando o usuário interagir com a apresentação, todos os componentes responderão à solicitação. Por exemplo: se uma interação "congelar" ocorrer, todas as cadeias de dados da apresentação deverão ser congeladas.

A figura 5.6 apresenta a distribuição das cadeias de dados da apresentação multimídia em termos de intervalos temporais. Essa visão permite simplesmente que se tenha uma noção do seqüenciamento da apresentação e não faz parte da especificação formal pelo modelo XHMBS.

A especificação apresentada a seguir contém somente os *hypercharts* relativos ao hiperdocumento, sem as tabelas de mapeamento, como apresentado na seção 5.4. Como o requisito quatorze da apresentação multimídia exige granularidade de milissegundos para a especificação do "jitter", adota-se a velocidade do *hyperchart* como 1 / ms, ou seja, os passos ocorrem a cada milissegundo. Esse mesmo requisito exige que as cadeias de áudio e vídeo

sejam sincronizadas a cada terço de segundo, de forma que uma SV, por exemplo, deve ser representada por quinze estados, sendo que cada estado representa $\frac{1}{5}$ de segundo (333 milissegundos). O fato de que deve haver sincronização a cada terço de segundo entre áudio e vídeo faz com que as cadeias de áudio sejam representadas exatamente na mesma granularidade das de vídeo, demandando, portanto, trinta estados para cada SA. Considerando que a apresentação deve ser especificada em duas direções, tem-se que para cada estado que representa a apresentação para frente, deve haver um estado que representa a apresentação para trás. Assim, considerando somente a especificação das cadeias de áudio e vídeo da apresentação multimídia tem-se cento e oitenta estados representando as seis SVs e outros cento e oitenta representado as SAs, sugerindo, assim, que a abordagem de especificação se baseie na notação parametrizada dos *hypercharts*. A figura 5.7 apresenta a especificação no contexto do hiperdocumento. Note-se que as interações “saltar”, “alterar a velocidade da apresentação”, “iniciar” e “terminar” são especificadas nesse nível, sendo disponíveis para todo o hiperdocumento. As interações “iniciar” e “terminar” são especificadas também para o contexto da apresentação multimídia, no estado MM.

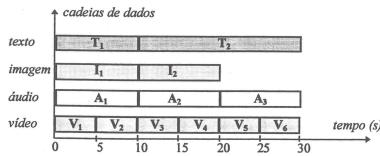


Figura 5.6 - Esquema da apresentação multimídia em uma linha de tempo

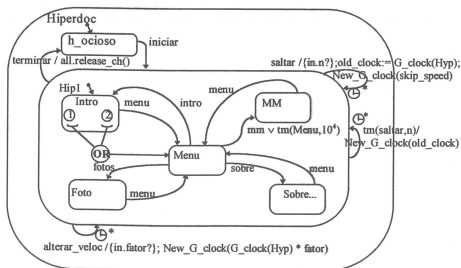


Figura 5.7 - Especificação do hiperdocumento em alto nível

Basicamente, os estados *Menu* e *Sobre...* já especificam no primeiro nível a semântica correspondente no domínio do hiperdocumento, ou seja, eles são estados básicos que podem ser mapeados para as páginas de texto que contêm as informações relativas, respectivamente,

ao menu de opções e ao autor do hiperdocumento. Os estados *Intro*, *Foto* e *MM* ainda devem ser decompostos. As figuras 5.8 e 5.9 especificam as decomposições, respectivamente, dos estados *Foto* e *Intro*.

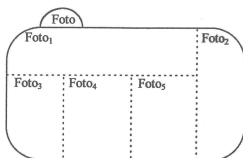
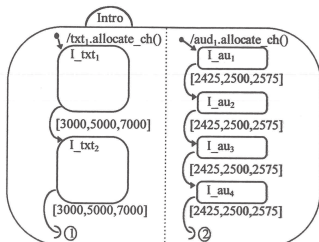


Figura 5.8 - Decomposição do estado *Foto* (galeria de fotos)

Na figura 5.8 os estados *Foto1*, *Foto2*, *Foto3*, *Foto4* e *Foto5* são especificados como a decomposição AND do estado *Foto*. Esses subestados são básicos e são mapeados para as páginas que contêm cada imagem. No estado *Intro* (figura 5.9) há um exemplo de que as alocações de canais são simplesmente requisitadas nas expressões de ação dos *hypercharts*, de forma que se considera que a abordagem de especificação é aquela em que os atributos dos canais são instanciados em tabelas à parte do *hyperchart*.



Os estados *I_txt1* e *I_aud4* receberam, respectivamente, as atividades: "On exit txt,release_ch()" e "On exit aud,release_ch()".

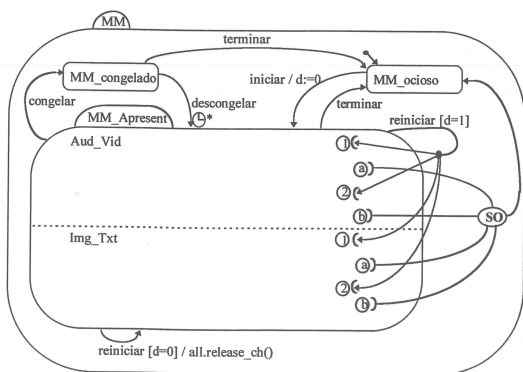
Figura 5.9 - Decomposição do estado *Intro*

Os requisitos de "jitter" especificados na decomposição do estado *Intro* são aqueles requeridos para a apresentação multimídia. Como se pode ver na figura 5.9, a especificação da pequena apresentação multimídia que introduz o hiperdocumento tem seus requisitos temporais especificados através de *transições temporais*. Além disso, a notação de *abstração de transições* é utilizada, representada pelos *pontos de conexão* 1 e 2, os quais determinam que

os arcos associados a eles são os *arcos* de origem da *transição M:N sincronizada* do tipo *or* especificada na figura 5.7.

A figura 5.10 apresenta o primeiro nível de abstração da decomposição do estado *MM*. Nessa figura, nota-se a especificação das interações “iniciar”, “terminar”, “congelar”, “descongelar” e “reiniciar”, aproximadamente de acordo com o bloco de construção sugerido na figura 5.5. A *transição M:N sincronizada* do tipo *strong-or* que tem origem nos subestados dos estados *Aud_Vid* e *Img_Txt* tem destino no estado *MM_ocioso* para satisfazer ao requisito dezoito da especificação textual da apresentação. Além disso, como se deseja que a interação “terminar” ocorra também quando a apresentação estiver congelada, criou-se a transição associada ao evento *terminar*, com origem no estado *MM_congelado*, além daquela que tem origem no estado *MM_Apresent*. Outro detalhe que difere da especificação do bloco de construção proposto é a ausência da ação *all.release_ch()* nas transições associadas ao evento *terminar*. Essa ação foi especificada em forma de atividade do tipo “*on entry*” no estado *MM_ocioso*. Isso garante que os canais serão liberados também quando o estado *MM_ocioso* for ativado pela *transição M:N sincronizada*, para a qual não é permitido o uso de expressões de ação.

Vale notar que a interação “reiniciar” é prevista para as duas direções de apresentação, modeladas pela variável *d*, conforme sugerido na figura 5.5. Para a interação na direção oposta à normal (*d = 1*), utilizou-se, também como proposto no bloco de construção, uma transição *M:N* convencional com origem no estado *MM_Apresent* e destino nos subestados finais das cadeias de dados da apresentação.



O estado *MM_ocioso* recebeu a atividade “*On entry all.release_ch()*”

Figura 5.10 - Primeiro nível de abstração da decomposição do estado *MM*

Na figura 5.11 o estado *Img_Txt* é decomposto. Essa decomposição representa dois blocos, cada um composto de um quadro de texto e um quadro de imagem, os quais são sincronizados por transições do tipo *master* controladas pelas restrições temporais das imagens. Nesse nível, tais restrições temporais não podem, ainda, ser observadas, mas como se pode observar pela figura 5.13, o estado I_1 representa os primeiros dez segundos de apresentação dessas cadeias, enquanto que os vinte segundos restantes são representados pelo estado I_2 .

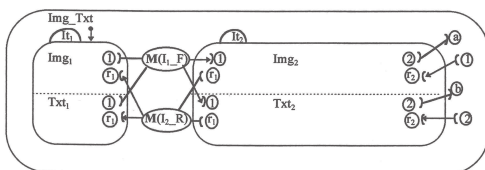


Figura 5.11 - Decomposição do estado *Img_Txt*

A figura 5.12 apresenta a decomposição do estado *Aud_Vid* da figura 5.10. Os noventa estados que especificam os blocos de áudio e vídeo sincronizados em ambas as direções de apresentação são especificados através da notação de *estados parametrizados*. Novamente, são utilizadas *abstrações de transições*, desta vez, combinadas com a notação parametrizada. A associação entre *pontos de conexão* de estados especificados em níveis diferentes de abstração pode ser notada na *reescrita de comportamento* do estado AV_{90} .

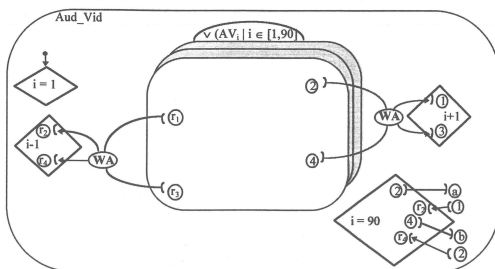


Figura 5.12 - Decomposição do estado *Aud_Vid*

O nível mais baixo de abstração da especificação é dado pelas decomposições dos estados Img_1 , Img_2 , Txt_1 e Txt_2 , além dos estados AV_i parametrizados. A figura 5.13 apresenta a decomposição dos estados Img_1 e Txt_1 , que são idênticas. Nessa figura, o bloco de

construção proposto na figura 5.4 é utilizado, novamente, com algumas modificações. A principal delas é a ausência de uma transição temporal partindo do estado I_{1_R} , para o caso do estado Img_1 , por exemplo. Isso significa que nenhuma restrição temporal é imposta sobre a apresentação em direção reversa da primeira imagem. A especificação textual da apresentação, não requer que o final da apresentação na direção oposta à normal ocorra de forma sincronizada.

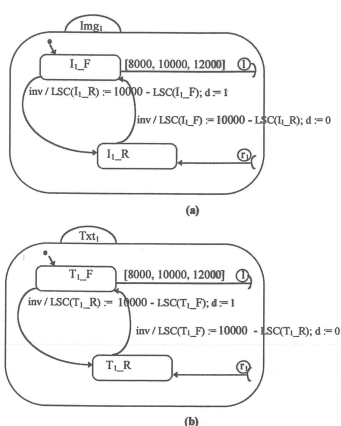


Figura 5.13 - Decomposição dos estados Img_1 e Txt_1

Na figura 5.14 o estado Img_2 é decomposto. Note-se a utilização dos estados $I_{2_Null_F}$ e $I_{2_Null_R}$, os quais devem ser associados à página nula e são utilizados para balancear temporalmente o cenário de sincronização composto pela segunda imagem (com dez segundos de duração) e o segundo texto (com vinte segundos de duração). O balanceamento temporal é aconselhável em toda especificação, principalmente naquelas que exigem a interação “inverter a direção da apresentação”, como é o caso deste exemplo. Dessa forma, a ocorrência de uma inversão após a apresentação da segunda imagem (associada aos estados I_{2_F} e I_{2_R}) é facilmente modelada como uma transição entre os estados $I_{2_Null_F}$ e $I_{2_Null_R}$.

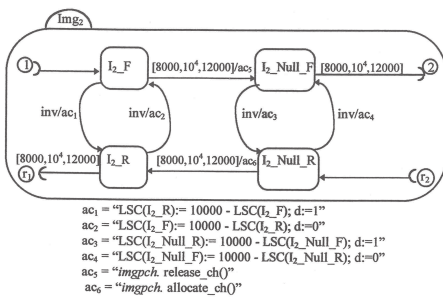


Figura 5.14 - Decomposição do estado Img_2

A figura 5.15 apresenta a decomposição do estado Txt_2 . Essa decomposição tem a forma sugerida pelo bloco de construção da figura 5.4.

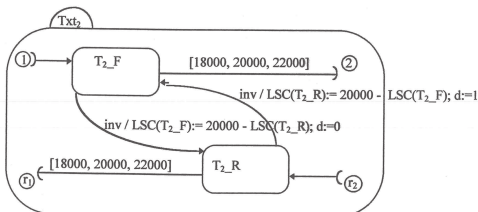


Figura 5.15 - Decomposição do estado Txt_2

A especificação se completa com a decomposição dos estados parametrizados AV_i , apresentada na figura 5.16, especificando o esquema de sincronização entre as páginas de áudio e vídeo. Por praticidade, utilizou-se o indicador de estado "default" para os estados A_i_F e V_i_F . No entanto, essas indicações de estados "default" só têm efeito para o estado AV_1 , quando da primeira ativação da apresentação. Os demais subestados do tipo A_i_F e V_i_F são ativados diretamente pelas transições às quais estão relacionados.

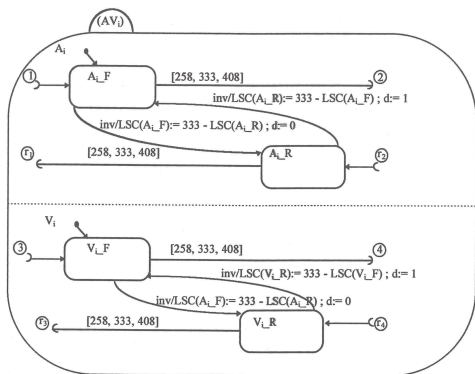


Figura 5.16 - Decomposição dos estados parametrizados Av_1

É importante notar que a interação de inverter a direção de apresentação é modelada por um único evento (*inv*). Dessa forma, não é necessária a utilização de qualquer mecanismo de re-sincronização além daquele imposto pela semântica de *hypercharts* para que toda a apresentação reaja à interação.

A próxima seção apresenta uma comparação entre a especificação desse exemplo através do modelo XHMBS e outros resultados obtidos com modelos baseados em redes de Petri.

5.6.3 Comparação de resultados

Em [Coo95], como dito anteriormente, a apresentação multimídia do exemplo especificado na seção 5.6.2 é especificada através de cinco modelos. Essas especificações permitiram que se obtivesse uma tabela comparativa checando a satisfação de alguns requisitos de aplicações multimídia classificados naquele trabalho, os quais podem ser interpretados em termos da classificação apresentada no capítulo 2 deste trabalho.

A comparação realizada entre os modelos também levou em consideração a complexidade da especificação, medida em termos da quantidade de símbolos utilizados para a construção, naquele caso, das redes de Petri. Os símbolos considerados foram blocos de construção (como os apresentados nas figura 3.10 e 3.13), lugares das redes de Petri convencionais, tanto subjacentes aos blocos de construção, quanto os utilizados da forma convencional, transições, arcos de entrada e saída e rótulos. Foram contadas todas as aparições de cada bloco de construção, mas os símbolos que compunham esse bloco só foram contados nos diagramas de decomposição em que apareceram. Assim, se tivessem sido utilizados cem

blocos de construção A , descritos pelos símbolos C , D e E , os quais tivessem aparecido somente em um diagrama de decomposição, o número de símbolos seria cento e três, relativo aos cem blocos A e aos blocos C , D e E .

A contagem dos símbolos da especificação em XHMBS foi realizada segundo dois princípios: contaram-se todos os símbolos utilizados, mas, para as parametrizações, cujas notações são suportadas pelos *hypercharts*, foram contados somente os símbolos constantes do respectivo *estado parametrizado*, não sendo multiplicada pela quantidade de estados definida no *intervalo de parametrização*; as *transições M:N sincronizadas* são símbolos atômicos em *hypercharts*, de forma que não se contam arcos de entrada e de saída como ocorre com as transições de redes de Petri. Os rótulos envolvidos, no entanto, são contados um a um, pois podem diferir em conteúdo; finalmente, não se consideraram as tabelas de mapeamentos do modelo XHMBS, procurando manter a abordagem no mesmo padrão adotado pelos modelos analisados em [Coo95], segundo os quais, os rótulos dos componentes da rede de Petri identificam os recursos (ou porções de informação) a que estes estão associados. Em outra abordagem, a contagem dos símbolos foi realizada conforme o que se fez com as especificações através de redes de Petri, ou seja, todos os arcos, identificadores de transições M:N, rótulos, estados, e outros símbolos atômicos foram contados como se a especificação não utilizasse *estados parametrizados*. As abstrações de transições foram consideradas e, portanto, diagramas que decompunham especificações realizadas em níveis mais altos de abstração foram contados somente uma vez.

A tabela 5.6 é uma extensão da tabela apresentada em [Coo95] em que os resultados para o modelo XHMBS estão contidos. Com pode-se notar, o XHMBS satisfaz a todos os requisitos e possui, quando utilizada a notação de *estados parametrizados*, complexidade da mesma ordem de grandeza do modelo $TSPN_{UI}$, que pode ser considerado equivalente em termos de poder de especificação. Considerando-se a especificação em sua forma expandida, pode-se notar que o número de símbolos é maior do que os utilizados pelo modelo $TSPN_{UI}$, possuindo complexidade equivalente à do modelo TSPN em ordem de grandeza. É certo que não é trivial estabelecer uma equivalência exata entre a complexidade de um modelo baseado em *statecharts* e outro baseado em redes de Petri, visto que as características dos grafos que representam as especificações são diferentes. O que se espera com esses números é fornecer uma comparação aproximada entre os modelos. A especificação do estado *MM_Apresent* é apresentada informalmente em notação expandida no apêndice B.

Requisitos	OCPN	DTPN	TSPN	XOCPN	$TSPN_{UI}$	XHMBS
Dados de origem pré-arranjadas	sim	sim	sim	sim	sim	sim
Dependência de tempo dos dados de origem	implícito	implícito	implícito	implícito	implícito	sim
Tipo de meio do dado de origem	sim	sim	sim	sim	sim	sim

Múltiplos locais de origem	sim	sim	sim	sim	sim	sim
Duração da apresentação	sim	sim	sim	sim	sim	sim
Local de destino único	sim	sim	sim	sim	sim	sim
Jitter	não	não	sim	não	sim	sim
Interação com o usuário	não	parcial	não	não	sim	sim
Nível de granularidade	muito alto	muito alto	adequado	adequado	adequado	adequado
Número total de símbolos	86	299	1117	19332	721	198 (parametrizado) 2696 (expandido)

Tabela 5.6 - Complexidade de alguns modelos de especificação

5.6.4 O modelo XHMBS e os requisitos de aplicações hipermídia

A exemplo do que foi feito com o HMBS na seção 3.4.4, nesta seção o XHMBS é analisado frente à classificação de requisitos de aplicações hipermídia apresentada no capítulo 2. Ainda da mesma forma que na análise do HMBS, os resultados dessa comparação são resumidos em uma tabela, facilitando o acesso a esses dados.

5.6.4.1 Requisitos relativos à obtenção e organização da informação

O XHMBS, como o HMBS, é um modelo voltado ao projeto das características navegacionais (ou comportamentais) de um hiperdocumento. Apesar das várias extensões realizadas, a análise da aplicação continua sendo um passo do processo de desenvolvimento que não é satisfeito pelo modelo, ou seja, para se desenvolver uma aplicação hipermídia não é suficiente a utilização de um modelo de hiperdocumentos. É necessário um processo mais abrangente que permita a identificação e a modelagem dos relacionamentos entre os elementos do domínio, o que, só então, deverá gerar os dados que serão úteis à fase de projeto. Dessa forma, pode-se dizer que os modelos HMBS e XHMBS podem ser vistos como fases de um processo de desenvolvimento mais abrangente que ainda deve ser definido.

Modelagem do domínio da aplicação

A modelagem do domínio da aplicação pode somente ser apoiada pelo modelo no sentido de que é possível modelar a estrutura da aplicação com *hypercharts*, mesmo em um nível de abstração mais alto do que o de projeto. No entanto, mesmo essa utilização da técnica de *hypercharts* para a análise da aplicação deve ser acompanhada de algum processo global que permita a identificação dos elementos do domínio. Dessa forma, a rigor, considera-se que o XHMBS não satisfaz ao requisito de modelagem do domínio da aplicação.

Derivação de elementos da aplicação

De forma semelhante ao que ocorre com a modelagem do domínio, pode-se dizer que o modelo XHMBS está em um nível de abstração no processo de desenvolvimento das aplicações em que características como o suporte à derivação de elementos da aplicação já devem ter sido providos. Assim, esse requisito também não é satisfeito.

Estruturação da informação

O XHMBS, da mesma forma que o HMBS, suporta a estruturação da informação nas formas linear, hierárquica e em rede. Adicionalmente, as características de especificação temporal incorporadas ao modelo pelos *hypercharts* incrementam a capacidade de estruturação do modelo.

Separação entre estrutura e conteúdo da informação

Esse requisito é satisfeito completamente pelo modelo XHMBS, o que não ocorre com o modelo HMBS. A definição de âncoras e o uso de mapeamentos estados-páginas e eventos-âncoras distintos faz com que toda a estrutura esteja em um nível diferente daquele em que se encontram as porções de conteúdo da informação, como proposto pelo modelo Dexter de referência para hipertextos.

Adicionalmente, a forma com que os mapeamentos são definidos no modelo XHMBS permite maior flexibilidade na especificação, possibilitando que elementos, tanto da estrutura quanto do conteúdo da informação, sejam utilizados de formas semanticamente diferentes como, por exemplo, que um mesmo evento esteja associado a várias âncoras, dependendo do contexto de informação ou das versões que estejam sendo utilizadas.

Várias visões da mesma informação

O XHMBS satisfaz esse requisito permitindo que várias páginas sejam associadas a um mesmo estado e que várias âncoras sejam associadas a um mesmo evento de uma certa transição. A multiplicidade dessas associações dá origem a versões do hiperdocumento, as quais podem ser selecionadas em tempo de navegação através dos *pchannels*, de acordo com as necessidades do usuário. Uma restrição imposta pelo modelo é que todas as páginas associadas a um mesmo estado devem ser associadas a um mesmo *pchannel*, possibilitando que somente uma das versões seja apresentada por esse canal a cada instante. Isso faz com que todas as páginas que compõem versões de um hiperdocumento devam ser do mesmo tipo de mídia. Esse problema, porém, pode ser resolvido através da estruturação de estados do *hyperchart* e de ligações convencionais do hiperdocumento.

Versões "sob medida" do mesmo hiperdocumento

Como ocorre com o HMBS, é possível que se definam, para o mesmo hiperdocumento, níveis diferentes de visibilidade de acordo com cada usuário. Assim se um CD-ROM for produzido para um determinado documento, por exemplo, poderão ser feitas algumas cópias com um certo nível de visibilidade e outras cópias com níveis de visibilidade diferentes do primeiro. Além disso, o uso de variáveis e condições que testam seus valores nos rótulos das transições do *hyperchart* constitui o suporte para mecanismos de controle de acesso no

hiperdocumento. Se houver, por exemplo, páginas nas quais são executadas aplicações que realizam algum tipo de intercâmbio de dados (senhas, preferências, etc), pode-se utilizar os dados obtidos para definir, em tempo de navegação, as regiões do hiperdocumento que deverão estar disponíveis para um determinado usuário.

Especificação em vários níveis de abstração e de granularidade

Os *hypercharts* permitem que vários níveis de abstração sejam utilizados, possibilitando que a abordagem estruturada de especificação seja utilizada. Conforme os níveis de abstração aumentam, pode-se explorar níveis diferentes de granularidade, tanto para a especificação da estrutura dos estados, quanto da aplicação de mecanismos de sincronização. Como ocorre com o HMBS, esse requisito é satisfeito de acordo com as necessidades do especificador. O uso de blocos de construção como pontos de partida para a modelagem dos padrões de sincronização e estruturação é uma prática que pode ser bastante útil quando grandes hiperdocumentos devem ser especificados.

5.6.4.2 Requisitos relativos à sincronização da apresentação

Esse grupo de requisitos constitui o foco principal do modelo XHMBS. A disponibilização de uma linguagem de alto nível com capacidade de especificação temporal e a incorporação de atributos específicos aos *pchannels* e *channels* são os principais esforços para a satisfação dos requisitos de sincronização.

Local de origem e Local de destino

Os locais de origem podem ser especificados através dos *pchannels* e *channels*, os quais permitem determinar se o local de origem é o próprio equipamento que apresenta o hiperdocumento ou se é um equipamento remoto.

A abordagem do modelo XHMBS considera que o local de destino da aplicação é aquele em que a especificação é executada. Não são suportados vários locais de destino para o mesmo hiperdocumento. Essa restrição, no entanto, pode ser resolvida com a associação de um *channel* a um atributo *destino* dos *pchannels* relacionados, fazendo com que esse *channel* funcione como a entidade de rede que cuida da sincronização da apresentação distribuída da informação em vários locais de destino.

Intervalos de apresentação

A exemplo do que fazem os modelos baseados em redes de Petri com notação temporal, a figura 5.17 apresenta a modelagem, através de *hypercharts*, das relações entre intervalos temporais propostas por Allen, mostrando que é possível modelar os intervalos de apresentação com XHMBS. A característica hierárquica dos *hypercharts* garante a possibilidade de aplicação recursiva desses relacionamentos, de forma que qualquer cenário de sincronização complexo pode ser modelado através de vários níveis de abstração, em termos das sete formas básicas apresentadas na figura 5.17.

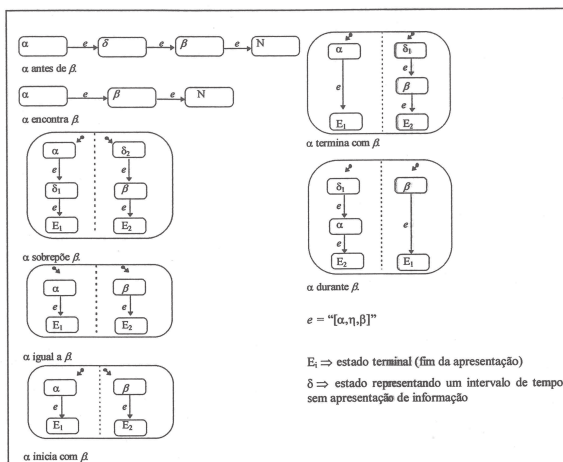


Figura 5.17 - Representações com *hypercharts* dos relacionamentos temporais de Allen

Tipos de fonte

Os *pchannels* do XHMBS permitem especificar se a fonte de uma determinada cadeia de dados é de tempo real ou pré-orquestrada. No entanto, não são determinadas formas diferentes de tratamento de cada um desses tipos de fonte. Pode-se dizer que quando a fonte é de tempo real e, portanto, não se conhece o conteúdo da informação, a especificação em *hypercharts* subjacente ao hiperdocumento simplesmente determina a preempção da apresentação desses dados, garantindo somente sua correção temporal, sem qualquer garantia com relação a sua semântica.

Dependência temporal

Esse requisito é implicitamente satisfeito. A especificação temporal determina a continuidade da apresentação das cadeias de dados, sejam discretas ou contínuas. Adicionalmente, não se impõe restrições à forma como a modelagem é realizada e, portanto as páginas podem conter dados contínuos, cujas apresentações são garantidas pelos respectivos *pchannels*, ou discretos, os quais podem, ainda, ser apresentados por algum tempo.

Alguns atributos dos *pchannels* e *cchannels* indicam a prioridade de tratamento das cadeias de dados, como sensibilidade a erros e outros. Dessa forma, através desses atributos, pode-se especificar tratamentos distintos a cadeias contínuas ou discretas.

Atraso

Além dos intervalos de apresentação definidos entre estados através de *transições temporais*, é possível especificar através de *cchannels* o atraso de rede tolerável para certas cadeias de dados. Assim, esse requisito pode ser completamente especificado.

“Jitter”

Os *eventos temporais* da forma $[\alpha, \eta, \beta]$ são dedicados à especificação do atraso (intervalo de tempo) entre a apresentação de duas páginas (η) e à variação desse intervalo de tempo (α e β). Como ocorre com os modelos TSPN, TSPN_{UI} e HTSPN, esse tipo de expressão temporal garante a especificação do “jitter”.

Taxas de “throughput”

As taxas de “throughput” podem ser especificadas através dos canais. O XHMBS não se preocupa com os detalhes dessa especificação, de forma que as taxas de “throughput” devem ser interpretadas pelos sistemas de apresentação e comunicação para atingir o resultado exigido pela especificação.

Sensibilidade a erros

A sensibilidade a erros é especificada através de dois atributos específicos para esse fim nos *cchannels*: um para erros de bits e outro para erros de pacotes. Como ocorre com as taxas de “throughput”, essa especificação deve ser interpretada pelo sistema de apresentação para que os requisitos da aplicação sejam satisfeitos.

5.6.4.3 Requisitos relativos à interação entre usuário e computador

A modelagem da interação entre usuário e computador é realizada através da combinação entre as ligações definidas pelas transições dos *hypercharts* e os *pchannels*. Além da navegação pelo hiperdocumento, são disponibilizadas as seguintes interações de multimídia: “iniciar”, “terminar”, “reiniciar”, “inverter a direção de apresentação”, “congelar”, “descongelar”, “saltar” e “alterar a velocidade da apresentação”. Todas essas interações são disponibilizadas como âncoras ao usuário conforme o contexto do hiperdocumento durante a navegação. As operações “alterar a velocidade da apresentação” e “saltar” são de escopo global, pois alteram as propriedades do relógio global do sistema, constituindo uma restrição do modelo, a qual pode ser solucionada com a definição de *relógios contextuais*, cujas propriedades seriam válidas somente para determinadas porções do hiperdocumento (equivalentes a subhierarquias do *hyperchart*). Uma vantagem do modelo XHMBS com relação a outros modelos como o DTPN e o TSPN_{UI}, que também tratam da interação com o usuário, é que não são necessários algoritmos de re-sincronização ou lógicas que vão além da técnica subjacente para que as operações sejam realizadas: o próprio *hyperchart* garante a semântica das interações.

5.6.4.4 Requisitos relativos à apresentação da informação

Entre as extensões realizadas sobre o HMBS para se obter o XHMBS está a definição de *pchannels*, os quais permitem uma especificação mais completa da apresentação das informações do que aquela proporcionada pelos "readers" do HMBS.

Coordenação espacial

Os *pchannels* possuem atributos que permitem a especificação das regiões dos dispositivos de apresentação que serão ocupadas pelas páginas ou âncoras do hiperdocumento. Além disso, é possível definir prioridades entre os *pchannels*, de forma que uma implementação possa suportar a sobreposição de canais ou gerar erros em casos de alocações simultâneas dos mesmos recursos dos dispositivos de apresentação.

Controle de atributos globais da apresentação

Da mesma forma, os *pchannels* possuem atributos que proporcionam o controle, inclusive em tempo de navegação, de alguns atributos globais de apresentação. Esses atributos dependem do tipo de mídia que estiver sendo utilizada.

Tipo de mídia da cadeia de dados

Ao associar um *pchannel* a um conjunto de páginas, o desenvolvedor estará especificando o tipo de mídia da informação contida nessas páginas. Cada *pchannel* possui um atributo que especifica o tipo de mídia que ele suporta, de forma que pode haver, para uma mesma mídia (imagem, por exemplo), vários *pchannels* diferentes, suportando padrões distintos de apresentação, como gif, jpeg, etc.

5.6.4.5 Requisitos relativos à navegação do hiperdocumento

A capacidade de especificação dos requisitos de navegação no modelo XHMBS foi enriquecida com a notação temporal. Dessa forma, o conceito de hiperídia ativa, em que os hiperdocumentos possuem ligações temporais, passa a ser satisfeito.

Noção de visita a um nó e Ligações temporais

Um nó, representado em XHMBS por um estado do *hyperchart*, pode ser visitado pelo usuário segundo duas possibilidades. A visita pode ter seu tempo determinado unicamente pelo usuário, caracterizando uma ligação convencional do HMBS, ou, se após um determinado tempo, o usuário não seguir qualquer ligação, uma das ligações é seguida automaticamente, levando a outro nó. Esse último tipo de ligação é conhecido como ligação temporal e constitui o suporte ao conceito de hiperídia ativa no XHMBS.

Estruturas alternativas de acesso à informação

A possibilidade de acesso direto às páginas através de índices do HMBS é herdada pelo XHMBS, de forma que essa estrutura alternativa de acesso à informação é suportada pelo modelo. Uma melhoria encontrada no XHMBS é a definição de páginas, constituídas de dois

elementos distintos, o conteúdo e o título. Essa abordagem de definição constitui um fator facilitador de implementações de sistemas que suportem índices em XHMBS.

Ligações dinâmicas

Não são suportadas ligações dinâmicas no XHMBS propriamente, mas é possível que sistemas de apoio ao modelo se beneficiem dos títulos das páginas para a geração de tais ligações como se fossem índices por assunto, gerados durante a navegação.

Ligações dentro de componentes

O mecanismo de *história temporal* dos *hypercharts*, juntamente com as operações de modificação dos *relógios locais* dos estados permite que sejam definidas ligações dentro de componentes, no sentido que uma ligação pode modificar a referência temporal da apresentação da informação representada por um estado.

Contextos de informação

Como ocorre com o HMBS, é possível utilizar superestados do *hyperchart* para definir contextos de informação e ligações entre contextos diferentes. O uso da hierarquia tanto dos *statecharts* no modelo HMBS, quando dos *hypercharts* no modelo XHMBS, constitui uma técnica bastante poderosa de especificação.

Visão da hierarquia do hiperdocumento

Em termos da visualização da hierarquia do hiperdocumento, o XHMBS herda a definição do nível de visibilidade do HMBS assim como as operações de visualização hierárquica. Como dito anteriormente, não há incompatibilidade semântica entre os modelos nesse aspecto.

Ainda não se sabe, no entanto, quais são os efeitos da utilização de visões hierárquicas em apresentações multimídia, visto que podem ocorrer intrusões de informação, as quais podem prejudicar a semântica da apresentação. A utilização do mecanismo de visões hierárquicas pode ser benéfica, no entanto, se a especificação do hiperdocumento for realizada com bom-senso, de forma controlada.

Um resumo dessa discussão sobre o modelo XHMBS é apresentado na tabela 5.6.

Requisito	Satisfeito pelo XHMBS	Observação
Modelagem do domínio da aplicação	√	O modelo XHMBS é um modelo de projeto de hiperdocumentos. No entanto esse requisito pode ser parcialmente satisfeito se os <i>hypercharts</i> forem utilizados para a análise da aplicação.

Requisito	Satisfeito pelo XHMBS	Observação
<i>Derivação de elementos da aplicação</i>	×	O XHMBS não suporta esse requisito. Presume-se que um passo anterior, em um processo mais abrangente, seja capaz de suportar a derivação automática de elementos da aplicação.
<i>Estruturação da Informação</i>	√	Da mesma forma que o HMBS, é possível estruturar a informação de forma linear, hierárquica ou em rede.
<i>Separação entre estrutura e conteúdo da informação</i>	√	Os mapeamentos entre estados e páginas e entre eventos e âncoras, além das restrições impostas sobre eles, permitem que a estrutura e o conteúdo da informação sejam especificados separadamente.
<i>Várias visões da mesma informação</i>	√	É possível que várias páginas sejam associadas a um mesmo estado e que várias âncoras sejam associadas a um mesmo evento. Juntamente com as restrições impostas a essas associações é possível obter várias versões do mesmo hiperdocumento disponíveis em tempo de navegação.
<i>Versões "sob medida" do mesmo hiperdocumento</i>	√	Assim como no HMBS, é possível que se tenha níveis de visibilidade diferentes para versões diferentes do mesmo hiperdocumento. O controle de acesso através de variáveis também é herdado do HMBS
<i>Especificação em vários níveis de abstração</i>	√	Os <i>hypercharts</i> , assim como os <i>statecharts</i> , permitem a decomposição de estados, suportando a abordagem estruturada de especificação.
<i>Especificação em vários níveis de granularidade</i>	√	Conforme aumentam os níveis de abstração de uma especificação, pode-se especificar a estruturação e a aplicação da sincronização em níveis igualmente diferentes de granularidade.
<i>Local de origem da cadeia de dados</i>	√	Essa especificação é suportada por atributos dos <i>pchannels</i> e <i>cchannels</i> .
<i>Local de destino da cadeia de dados</i>	√	A abordagem do XHMBS considera que o local de destino de uma aplicação é a máquina que executa a especificação. Não são suportados vários locais de destino.
<i>Intervalos de apresentação</i>	√	A notação temporal provida pelas <i>transições temporais</i> garante a modelagem por XHMBS das relações temporais de Allen. A hierarquia dos <i>hypercharts</i> permite a aplicação recursiva desses relacionamentos, tornando o modelo completo.
<i>Tipos de fonte</i>	√	Atributos dos canais de apresentação e de comunicação permitem que se especifique o tipo da fonte suportada pelo canal e qual deve ser o tratamento para os dados que

Requisito	Satisfeito pelo XHMBS	Observação
		serão apresentados ou transportados pela rede.
<i>Dependência temporal</i>	√	A satisfação desse requisito é implícita, visto que as <i>transições temporais</i> especificam o intervalo de apresentação da informação associada a um estado. Além disso, atributos dos canais permitem que sejam especificados tratamentos apropriados para dados discretos ou contínuos.
<i>Atraso</i>	√	Esse requisito é especificado em termos da apresentação pelas <i>transições temporais</i> e em termos da comunicação por um atributo dos <i>channels</i> .
"Jitter"	√	Satisfeito diretamente pelos <i>eventos temporais</i> .
<i>Taxas de "throughput"</i>	√	Esse requisito é satisfeito, mas a implementação da especificação deve considerar as características da plataforma para obter as taxas exigidas pela especificação.
<i>Sensibilidade a erros</i>	√	Os atributos dos <i>channels</i> permitem especificar a tolerância a erros de bits e de pacotes.
<i>Interação Usuário-Computador</i>	√	São suportadas as principais interações com o usuário encontradas em aplicações multimídia, além de navegação convencional em aplicações hipermídia.
<i>Coordenação espacial</i>	√	Existem atributos específicos para determinar as porções dos dispositivos de apresentação que devem ser ocupadas por um determinado <i>pchannel</i> , além das definições de prioridades que permitem o melhor gerenciamento dos recursos do sistema.
<i>Controle de atributos globais da apresentação</i>	√	Assim como para a coordenação espacial da apresentação, existem atributos que são globais à apresentação de todos os dados que são apresentados por um <i>pchannel</i> , os quais podem ser modificados em tempo de navegação.
<i>Tipo de mídia da cadeia de dados</i>	√	A associação entre uma página e um <i>pchannel</i> determina o tipo dos dados daquela cadeia. O problema com esse tipo de especificação é a impossibilidade de verificação da correção da associação em tempo de autoria, assim como ocorre com o HMBS.
<i>Noção de visita a um nó</i>	√	Suporta a noção convencional de visita, cujo tempo é determinado somente pelo usuário, e a noção de visita de hipermídia ativa, em que uma ligação temporal pode ser especificada.

Requisito	Satisfeito pelo XHMBS	Observação
Estruturas alternativas de acesso à informação	√	Os índices são suportados pelo XHMBS da mesma forma que pelo HMBS. A nova definição de páginas melhora o conceito de títulos, facilitando a implementação dos índices.
Ligações temporais	√	Transições temporais ou associadas ao evento de "time-out" permitem a modelagem desse tipo de ligação.
Ligações dinâmicas	×	Trata-se de uma característica que não é pertinente ao modelo, mas a um sistema que o suporte. A definição de páginas, com títulos e conteúdo formalmente separados pode facilitar a implementação de índices por assunto.
Ligações dentro de componentes	√	Esse requisito é suportado pela utilização conjunta do mecanismo de <i>história temporal</i> e de operações de modificação dos <i>relógios locais</i> dos estados.
Contextos de informação	√	Esse requisito é suportado pela utilização de superestados como elementos de definição de contextos do hiperdocumento.
Visão da hierarquia do hiperdocumento	√	O mecanismo de visões hierárquicas permite a satisfação desse requisito. Algum cuidado adicional deve ser tomado quanto à utilização de hierarquia com dados multimídia.

Tabela 5.6 - XHMBS e os requisitos de aplicações hiperfídia

5.7 Implementação de especificações XHMBS

Basicamente, o principal elemento de uma especificação em XHMBS é o *hyperchart* subjacente ao hiperdocumento. Dessa forma, o principal aspecto a se considerar para a implementação de uma especificação em XHMBS é a forma de se implementar a execução desse *hyperchart*. Grosso modo, pode-se dizer que há duas formas de se implementar um *hyperchart* e, assim, uma especificação em XHMBS: a geração da aplicação em formato consumível por algum software de apresentação e a interpretação do *hyperchart* em um ambiente específico, a partir da qual, conforme os mapeamentos, os dados serão apresentados exatamente como especificado na semântica de navegação do modelo.

No primeiro caso devem ser utilizados algoritmos que permitam que determinados padrões de especificação em *hypercharts* sejam transformados em código-fonte de alguma linguagem, a qual pode ser compilada ou interpretada por aplicativos apropriados para a apresentação de aplicações hiperfídia. Assim, somente um ambiente de edição de *hypercharts* pode ser suficiente para a realização desse tipo de implementação. No segundo caso é necessário que exista um ambiente capaz de executar especificações em XHMBS, ou seja, apresentar os hiperdocumentos a partir da execução de um *hyperchart*. A principal característica de um ambiente desse tipo deve ser a existência de uma máquina de *hypercharts*, ou seja, um conjunto de procedimentos que executem, de forma interpretada, uma

especificação realizada com *hypercharts*. Em outras palavras, é necessário um simulador de *hypercharts* que tenha seu funcionamento associado a um sistema de apresentação, o qual implementa, na realidade, o conjunto de *pchannels* e *cchannels* disponíveis para o ambiente. As subseções seguintes apresentam algumas possibilidades para a viabilização de ambos os tipos de suporte à implementação de especificações em XHMBS.

5.7.1 Suporte traduzido ao XHMBS

A tradução da especificação em algum tipo de código-fonte vem sendo explorada em alguns experimentos em que os tipos de sincronização das *transições M:N sincronizadas* dos *hypercharts* são implementados sobre o software "MacroMedia Director[®]", voltado para a composição de aplicações hiperídia. O objetivo desse trabalho é o de reunir informações que permitam a elaboração de um conjunto de diretrizes que racionalizem o processo de implementação de especificações XHMBS através desse software.

A composição de apresentações no "Director[®]" é realizada em função de uma linha de tempo e de canais. Essas duas dimensões dão origem ao conceito de "frames", que são representados por projeções sobre a linha de tempo, relacionando, assim, todos os canais a cada instante da apresentação, como mostra a figura 5.18. Um "frame" suporta, além da informação (I) que será apresentada, a inclusão de "scripts" (S) que podem exercer ações de controle sobre a apresentação, como posicioná-la em pontos específicos da linha de tempo ou atualizar o valor de variáveis de implementação.

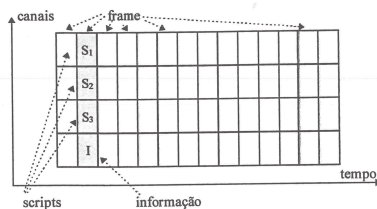


Figura 5.18 - Visão esquemática do ambiente de composição do "Director[®]"

A abordagem utilizada para a implementação de especificações em XHMBS é a colocação das páginas nos canais de apresentação de informação apropriados, como se esses correspondessem aos *pchannels*. Além disso, utilizam-se "scripts" que implementam o comportamento dos *hypercharts*, atualizando valores de variáveis de estado. Assim, unindo uma forma padronizada de disposição de frames sobre a linha de tempo ao uso de "scripts" também padronizados em sua lógica, tem-se como resultados, até o momento, a execução de *transições temporais* e dos tipos *strong-or*, *weak-and* e *master* de *transições M:N sincronizadas* através do "Director[®]". Em [Hir97] essas implementações poderão ser vistas com mais detalhes. Outras possibilidades no mesmo sentido desse trabalho e que ainda não foram exploradas são a geração de código HTML para hipertextos e Java para aplicações

hipermídia mais complexas. Isso caracterizaria um passo na direção da implementação de aplicações hipermídia distribuídas sobre a WWW, até o momento não exploradas, visto que o "Director[®]" implementa somente aplicações locais.

5.7.2 Suporte interpretado ao XHMBS

Em termos da criação de um ambiente de apoio ao XHMBS, tem-se a perspectiva de extensão do ambiente HySChart ("Hypermedia System based on stateCharts"), atualmente em desenvolvimento no ICMSC-USP, o qual apóia a autoria e navegação de hiperdocumentos especificados em HMBS. O HySChart está sendo implementado como uma extensão do ambiente Statsim [Mas91] e, portanto, baseia-se em uma máquina de simulação de statecharts convencionais. Dessa forma, o trabalho de extensão do ambiente HySChart para apoiar o XHMBS deveria ser conduzido em duas frentes: a extensão do Statsim para a obtenção de uma máquina de *hypercharts*, e a extensão das características de autoria e navegação do próprio ambiente HySChart para o suporte aos recursos do XHMBS.

A extensão do ambiente Statsim para o suporte a *hypercharts* deve ser realizada tanto sobre o editor quanto sobre o simulador de statecharts. O editor deverá ser modificado para suportar os novos recursos de notação dos *hypercharts*, como as *transições M:N sincronizadas*, os *relógios locais*, o mecanismo de *história temporal* e as restrições sintáticas sobre a utilização de *eventos temporais*. O uso de ações e do mecanismo de história convencional, que não são definidos pelo statechart do HMBS já são suportados pelo ambiente Statsim e não precisam de modificações adicionais. No entanto, o suporte à notação de *estados parametrizados* constituiria uma grande melhoria ao ambiente, visto que o poder dessa notação é de crucial importância para a especificação de aplicações cuja característica multimídia seja a mais importante. Para tanto é necessário que se defina a semântica dessa notação, apresentada neste trabalho somente através de exemplos. As demais notações têm suas semânticas caracterizadas no capítulo 4.

O simulador do ambiente Statsim pode ser modificado de forma que suas rotinas executem o statechart subjacente e devolvam uma configuração do *hyperchart* ou então, de forma que o *hyperchart* inicial seja pré-interpretado, gerando um statechart, o qual poderá ser executado segundo as rotinas atuais do simulador, salvo as modificações relativas aos *relógios locais* e *registradores de história temporal* que devem, de qualquer maneira, ser suportadas.

Além da máquina de statecharts, o subsistema de autoria e navegação de hiperdocumentos do HySChart deveria sofrer algumas alterações, principalmente para suportar os novos mapeamentos propostos pelo XHMBS. Mais do que isso, deveriam ser implementadas algumas classes de *pchannels* e *cchannels*, assim como seus mecanismos de instanciação. Um detalhe importante relativo à criação de um simulador de *hypercharts* baseado no ambiente Statsim é a sua relação com o ambiente de navegação de hiperdocumentos, no sentido de que as páginas, segundo o XHMBS, são associadas aos estados do *hyperchart* e não do statechart subjacente, o qual deverá ser sempre transparente à aplicação. Isso pode ser facilmente resolvido fazendo com que as configurações fornecidas pelo simulador ao navegador sejam relativas somente aos estados do *hyperchart*.

Como um passo inicial na direção da criação de um ambiente de apoio ao XHMBS, desenvolveu-se um modelo de objetos procurando estender a especificação realizada para o

ambiente HySChart. Esse modelo de objetos, desenvolvido segundo o método Fusion de desenvolvimento orientado a objetos [Co194], é apresentado no apêndice A deste trabalho.

5.8 Considerações finais

Este capítulo apresenta o modelo XHMBS, com suas novas características, as quais aumentam o seu poder de especificação com relação ao HMBS. As principais extensões realizadas são aquelas relacionadas com a satisfação dos requisitos de sincronização, principalmente a especificação temporal e a sincronização entre cadeias de dados. Além dessas extensões, de maior impacto, mostrou-se que é possível dotar o modelo de notações que o tornam capaz de especificar requisitos relativos à apresentação da informação e de interação com o usuário.

As maiores diferenças entre o modelo XHMBS e seu antecessor, o HMBS, são a utilização de *hypercharts* pelo primeiro em vez de *statecharts* pelo segundo, além das definições de mapeamentos entre estrutura e conteúdo da informação, os quais permitem maior separação entre essas duas características da aplicação. A idéia de utilizar *pchannels* e *cchannels* como objetos supridos por um ambiente de apoio garante às implementações interpretadas de hiperdocumentos especificados em XHMBS maior flexibilidade em termos de interação com o usuário, já que, seguidas as diretrizes que determinam as características básicas dos canais, os mesmos podem ser implementados de diferentes maneiras, tanto sob o ponto de vista de software quanto de hardware.

Um exemplo da literatura foi especificado segundo o modelo XHMBS. A comparação dos resultados dessa especificação com os obtidos com especificações por outros modelos mostra que o XHMBS atende aos mesmos requisitos da aplicação e que, além disso, a complexidade da especificação gerada com esse modelo é menor do que a média das demais.

Finalmente, na seção 5.7, apresenta-se uma visão geral das duas formas principais de implementação de especificações XHMBS: a tradução e a interpretação, por assim dizer. No sentido de se obter um ambiente de apoio ao XHMBS, como ocorre atualmente com o HMBS, propõe-se um modelo de objetos que estende as características do ambiente HySChart. Esse modelo de objetos, apresentado no apêndice A, constitui o passo inicial para a criação de um ambiente de apoio interpretado ao modelo XHMBS.

Capítulo 6

Conclusões

6.1 Considerações finais

Este trabalho apresenta o modelo XHMBS, uma extensão do modelo HMBS que tem como principal característica a capacidade de especificação de requisitos temporais e de sincronização de aplicações hiperfídia. Além disso, o modelo também suporta a especificação de requisitos relativos à interação com o usuário e à coordenação espacial da informação. Dessa forma, o domínio de aplicações às quais o modelo XHMBS se aplica torna-se mais amplo do que aquele ao qual aplica-se o modelo HMBS.

Comprovou-se que a capacidade de especificação concorrente dos statecharts constitui a base conceitual sobre a qual é possível definir mecanismos de sincronização mais complexos. Além disso, com o uso do conjunto completo das características de statecharts, foi possível a definição de uma linguagem de mais alto nível, os *hypercharts*, através da qual a especificação dos requisitos temporais e de sincronização de hiperdocumentos complexos pode ser realizada de forma clara e concisa. O uso dessa linguagem como técnica subjacente ao modelo XHMBS tem dois efeitos principais: o primeiro é o ocultamento de especificações bastante complexas em statecharts e o segundo é sua compatibilidade com o modelo HMBS, uma vez que os *hypercharts* podem ser considerados um superconjunto de statecharts, possibilitando que a especificação navegacional de hipertextos convencionais seja realizada da mesma forma com qualquer um dos modelos.

As características dos canais, tanto de apresentação (*pchannels*) quanto de comunicação (*channels*), foram apresentadas brevemente e informalmente neste trabalho. A especificação da coordenação espacial de aplicações hiperfídia é extremamente importante para os modelos de especificação, uma vez que a melhor satisfação desse requisito pode constituir a base para modelos capazes de especificar aplicações muito mais sofisticadas, como aquelas baseadas em realidade virtual. Dessa forma, considera-se que os recursos notacionais apresentados neste trabalho são somente a conceitualização de um recurso de especificação que merece atenção e que deve ser mais explorado, complementando o modelo XHMBS nesse aspecto.

Como observado no primeiro capítulo deste trabalho, as soluções propostas na literatura foram de fundamental importância para este trabalho. Ressalta-se, neste ponto, a influência dos modelos TSPN e TSPN₀, que originaram a notação de *transições temporais* e *transições M:N sincronizadas*, e do modelo Amsterdam de hiperfídia, que deu origem à utilização dos canais. Como previsto, os modelos baseados em Redes de Petri constituíram um fator facilitador para o trabalho de extensão do modelo HMBS devido à similaridade entre essa técnica e os statecharts.

Assim, pode-se dizer que, diante dos resultados obtidos com o esforço de pesquisa que resultou nesta dissertação, foi possível mostrar uma forma de especificar aplicações hiperfídia com requisitos temporais e de sincronização complexos através de um modelo baseado no HMBS. No entanto, próximos trabalhos devem ser conduzidos com o objetivo de se realizar

melhorias e conduzir experimentos em torno do modelo XHMBS para que seja mostrada a utilidade prática do modelo.

6.2 Contribuições deste trabalho

Este trabalho constitui uma proposta original naquilo que se refere à especificação de requisitos temporais e de sincronização de informações multimídia baseada em statecharts. A sua principal contribuição é a proposta de uma linguagem baseada em statecharts convencionais especialmente dedicada à especificação de hiperdocumentos, os *hypercharts*. A adaptação para statecharts de conceitos de especificação propostos por outros modelos, muitos deles baseados em técnicas distintas de statecharts, oferece a uma comunidade de especificadores, a dos que utilizam máquinas de estados como técnica de especificação, um modelo compatível com seus conhecimentos e com um domínio de aplicações cuja demanda vem crescendo constantemente.

A forma de especificação de interações com o usuário através de *hypercharts* pode ser vista como uma vantagem do XHMBS em relação a alguns modelos baseados em redes de Petri apresentados neste trabalho, os quais caracterizam-se por permitir a especificação desse requisito. Nesses modelos, a saber: DTPN e TSPN_{UI}, utilizam-se complexas políticas de resincronização, como algoritmos escondidos sob a técnica, para que ocorra a apresentação correta dos dados. Com *hypercharts*, a simples utilização de transições com condições e ações especiais em seus rótulos permite a especificação das interações primitivas. É certo que, para o uso de interações compostas, ainda é necessário que a interface com o usuário pré-processe as requisições e gere os eventos necessários. Da mesma forma que o TSPN_{UI}, o XHMBS torna disponível um conjunto de interações com o usuário, as quais podem ser combinadas para gerar outros efeitos sobre a apresentação da informação. Por outro lado, essas interações não são parte integrante da linguagem de especificação e, assim, podem ser modificadas de acordo com a aplicação na forma de blocos de construção.

A introdução de conceitos de especificação de coordenação espacial da informação pode ser considerada um passo inicial na direção de outras melhorias do XHMBS, para a especificação de diversos tipos de aplicações, como jogos interativos e realidade virtual. Uma possibilidade relacionada com a realidade virtual e factível através do modelo XHMBS é a especificação da navegação por um ambiente virtual no qual podem ser encontradas informações. A questão é que em ambientes virtuais, a visualização da informação depende da presença do usuário e, além disso, do seu ponto de vista, ou seja, se o usuário olhar para o que deveria ser a sua frente, a informação é apresentada de uma forma. Se ele olhar para um de seus lados, essa mesma informação (por exemplo, a decoração de sua futura sala de estar) será apresentada de uma forma diferente. Essas várias visões da mesma informação podem ser representadas como versões da informação, ou seja, páginas mapeadas para um mesmo estado do *hyperchart*, apresentadas pelo mesmo *pchannel*. Assim, com a definição de uma classe de *pchannels* apropriada para a exibição de dados em realidade virtual, esse tipo de aplicação pode ser especificado.

O conceito de canais de comunicação (*cchannels*) também constitui uma inovação na forma de especificação do modelo XHMBS, uma vez que se determina que esses objetos possuem interface com os canais de apresentação (*pchannels*), enquanto que em outros modelos, como o OCPN, por exemplo, essa interface é estabelecida com o próprio modelo

comportamental subjacente. A abordagem do XHMBS parece ser mais conveniente, uma vez que a abstração de que um sistema de apresentação se comunica com a rede subjacente para obter os dados distribuídos é mais próxima de uma eventual implementação do hiperdocumento. Além disso, a união do conceito de *cchannel* com conhecimentos específicos na área de comunicação de dados pode dar origem a implementações desses objetos capazes de controlar as mais diversas formas de distribuição de aplicações hipermédia, principalmente aquelas em que são utilizados vários locais de destino distribuídos pela rede, não suportadas atualmente pelo XHMBS.

6.3 Sugestões de novas pesquisas

A partir das propostas apresentadas nos capítulos anteriores pode-se vislumbrar uma série de possibilidades no que se refere a extensões do modelo e novas pesquisas nessa área. Como discutido no capítulo 5, um caminho imediato a ser explorado é a implementação de um ambiente de apoio à autoria e navegação de aplicações hipermédia especificadas em XHMBS. Esse trabalho pode ter como ponto de partida a extensão do ambiente HySCharts, baseado no HMBS, atualmente em desenvolvimento no ICMSC-USP, e que se caracteriza atualmente como uma extensão do ambiente Statsim [Mas91]. Esse trabalho envolveria o suporte à notação de *hypercharts* e aos mapeamentos do XHMBS. A criação de um ambiente com essas características viabilizaria o que pode ser classificado como um suporte interpretado ao XHMBS, visto que o ambiente Statsim simula os statecharts, como se fosse um interpretador da linguagem. Nesse sentido, ressalta-se a importância de recursos notacionais de *hypercharts* que enriqueceriam grandemente o ambiente HySCharts, como o suporte à notação de *estados parametrizados*, o que permitiria que especificações arbitrariamente grandes fossem realizadas através de diagramas bastante concisos.

Por outro lado, pode-se optar por desenvolver trabalhos que visem a implementação de especificações XHMBS em plataformas específicas para a composição de aplicações hipermédia, como é o caso do trabalho que vem sendo realizado sobre o "DirectorSM" [Hir97] e de outras possibilidades com as linguagens Java e HTML. Esses trabalhos envolveriam, principalmente, a geração de código a partir de especificações em *hypercharts* e, conseqüentemente, statecharts.

Em qualquer uma dessas linhas de pesquisa, a definição de canais de apresentação e comunicação com características avançadas é um trabalho a parte que tem potencial para enriquecer profundamente as implementações de hiperdocumentos especificados em XHMBS. Especificamente, a definição de canais apropriados para a apresentação e transporte de dados de realidade virtual pode aumentar sensivelmente o escopo de aplicação do modelo.

A questão do uso da organização hierárquica da informação é um problema que merece grande atenção, principalmente quando se trata da apresentação simultânea de informações multimídia alocadas em níveis distintos de hierarquia do *hyperchart*. É necessário saber em que situações é interessante ter-se informações multimídia organizadas hierarquicamente e, além disso, é importante disponibilizar estratégias de implementação para esse tipo de organização da informação através de linguagens ou abordagens de implementação que organizam os dados somente de forma plana (linearmente ou em rede), como é o caso de softwares de composição como o "DirectorSM".

Como é possível perceber através dos exemplos apresentados no capítulo 5, a notação gráfica do modelo XHMBS, dada pelos *hypercharts*, tem seu foco voltado somente para a especificação comportamental do hiperdocumento, ficando a especificação dos demais mapeamentos restrita ao uso de tabelas, o que acaba por aumentar a complexidade da especificação. Dessa forma, uma possível melhoria do XHMBS pode ser a incorporação de elementos de informação (páginas e âncoras) à notação dos *hypercharts*, facilitando a visualização dos mapeamentos V_a e V_r . Além disso, é interessante que se ofereça uma visão espacial da especificação, ou seja, pode ser definida uma notação através da qual seja possível especificar graficamente a utilização dos *pchannels* e até mesmo dos *cchannels*, de forma que situações de conflito e erros relativos à utilização dos canais sejam facilmente visualizados, o que não é possível com as especificações em XHMBS sem apoio de validadores automatizados. Esse trabalho deveria ter como objetivo esconder a complexidade da especificação formal do modelo, provendo uma notação de uso mais simples. Essa simplificação da notação do modelo (syntaxe) poderia ser utilizado, para que um ambiente de composição de aplicações fosse implementado, escondendo do usuário, potencialmente leigo, os detalhes do formalismo.

Finalmente, a criação de um processo abrangente de especificação de aplicações hiperemídia é um trabalho que pode se beneficiar do XHMBS como modelo de projeto de hiperdocumentos. Esse processo, em linhas gerais, deveria definir formas de modelagem do domínio da informação, provendo os elementos de especificação consumíveis pelo XHMBS, como páginas, âncoras, ligações e versões, por exemplo. Em uma fase posterior à modelagem com XHMBS, esse processo poderia fornecer diretrizes de implementação e testes da especificação gerada, completando o ciclo de vida da aplicação.

Mais especificamente, no sentido da validação e testes de uma especificação realizada em XHMBS, podem ser exploradas adaptações para *hypercharts* de abordagens aplicadas tradicionalmente a statecharts, algumas delas já implementadas no ambiente Statsim, como árvore de alcançabilidade [Mas94], simulação programada [Can93], análise de hiperdocumentos baseada em statecharts [For96] e a aplicação do critério análise de mutantes a especificações realizadas com statecharts [Fab96]. Acredita-se que esses recursos de validação auxiliem o especificador na tarefa de identificar problemas relacionados com a especificação do hiperdocumento, subsidiando a melhoria da qualidade do produto final.

Bibliografia

- [Aks88] Akscyn, R. M., McCracken, D. L. & Yoder, E. A. – KMS: a distributed hypermedia system for managing knowledge organizations. *Communications of the ACM* 31(7): 820-835, July 1988.
- [All83] Allen, J. F. – Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11): 832-834, November 1983.
- [App94] Apple Computer – *Hypercard: stack design guidelines*. Addison - Wesley Publishing Company Inc., 1994.
- [Bee94] Beeck, M. van der – A comparison of statecharts variants. *Lecture Notes in Computer Science* 863: 128-148, September 1994.
- [Bro87] Brown, P.J. – Tuning ideas into products: the Guide system. *Proceedings of the ACM Hypertext' 87* (Chapel Hill, N C): 33 - 40, 1987.
- [Bus45] Bush, V. – As we may think. *The Atlantic* 176(1): 101-108, July 1945.
- [Can93] Cangussu, J. W. L. – *Execução programada de statecharts*. Dissertação de Mestrado, São Carlos - Brasil, ICMSC-USP, 1993.
- [Col94] Coleman, D. et al. – *Object oriented development: the Fusion method*. EUA, Prentice Hall object oriented series, 1994.
- [Coo95] Cooper, K. – *TSPN_{UI}: A Petri net model for specifying user interactions in multimedia presentations*. Master Thesis, Canadá, The University of British Columbia, 1995.
- [Dia93] Diaz, M. & Senac, P. – Time stream Petri nets: a model for multimedia streams synchronization. *Proceedings of the international conference on multi-media modelling*. Singapore, November 1993.
- [Fab96] Fabbri, S. C. P. F. – *A análise de mutantes no contexto de sistemas reativos: uma contribuição para o estabelecimento de estratégias de teste e validação*. Tese de Doutorado, São Carlos - Brasil, IFSC-USP, 1996.
- [For96] Fortes, R. P. M. – *Análise e avaliação de hiperdocumentos: uma abordagem baseada na representação estrutural*. Tese de Doutorado, São Carlos - Brasil, IFSC-USP, 1996.
- [Gar88] Garg, P. K. – Abstraction mechanisms in hypertext. *Communications of the ACM* 31(7): 862-870, 1988.

- [Gar93] Garzotto, F.; Paolini, P. & Schwabe, D. – HDM - a model-based approach to hypertext application design. *ACM Transactions on Information Systems* 11(1): 1-26, january 1995.
- [Gin95] Ginige, A.; Lowe, B. & Robertson J. – Hypermedia authoring. *IEEE multimedia* 2(4): 24-35, 1995.
- [Hal90] Halasz, F. & Schwartz, M. – The Dexter reference model. *Proceedings of the 1st Hypertext NIST Standardization Workshop*: 95-133, january 1990.
- [Har87a] Harel, D. – On the formal semantics of statecharts. *Proceedings of the 2nd IEEE Symposium on Logic in Computer Science*. Ithaca, New York: 4-64, 1987.
- [Har87b] Harel, D. – Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 8: 231-274, 1987.
- [Har94] Hardman, L.; Bulterman, D.C.A. & Rossum, G. van. – The Amsterdam Hypermedia Model. *Communications of the ACM* 37(2): 50-62, february 1994.
- [Hir97] Hiraoka, R. H. – *Relatório de Iniciação Científica*. São Carlos - Brasil, ICMSC-USP-CNPq (em preparação).
- [Isa95] Isakowitz, T.; Stohr, E.A. & Balasubramanian, P. – RMM: a methodology for structured hypermedia design. *Communications of the ACM* 38(8): 34-44, august 1995.
- [Lit90] Little, T. & Ghafoor, A. – Synchronization and storage for multimedia objects. *IEEE Journal on Selected Areas in Communications* 8(3): 413-427, april 1990.
- [Mas91] Masiero, P. C.; Fortes, R. P. M. & Batista Neto, J. do E. S. – Editando e simulando os aspectos comportamentais de sistemas de tempo real. *Anais do XVIII Seminário Integrado de Software e Hardware (SEMISH)*: 45-61, agosto 1991.
- [Mas94] Masiero, P. C.; Maldonado, J. C. & Boaventura, I. G. – A reachability tree for statecharts and analysis of some properties. *Information and Software Technology* 36(10): 615-624, 1994.
- [Mer74] Merlin, P. – *A study of the recoverability of computer systems*. Thesis, Irvine - EUA, Computer Science Dept.- University of California, 1974.
- [Oli95] Oliveira, M. C. F.; Turine, M. A. S. & Masiero, P. C. – *A statechart-based model for hypertext*. São Carlos - Brasil, Notas do ICMSC, Série Computação (19), 1995.
- [Pau96] Paulo, F. B.; Masiero, P. C. & Oliveira, M. C. F. – Extensões ao HMBS para especificação de apresentações hiperídia. *Anais do X Simpósio Brasileiro de Engenharia de Software*, São Carlos - Brasil: 241-256, 1996.
- [Pen96] Penteadó, R. A. D. – *Um método para engenharia reversa orientada a objetos*. Tese de doutorado, São Carlos - Brasil, IFSC-USP, 1996.
- [Pnu92] Pnuelli, A. & Kensten, Y. – Timed and hybrid statecharts and their textual representation. *Lecture Notes in Computer Science* 571: 591-620, january 1992.

- [Pra93] Prabhakaran, B. & Raghavan, S. V. – Synchronization models for multimedia presentation with user interaction. *ACM Multimedia 93*, California - EUA: 157-166, June 1993.
- [Ram74] Ramchandani, C. – *Analysis of asynchronous concurrent systems by timed Petri nets*. Master Thesis, EUA, MIT, February 1974.
- [Sen94] Sénac, P.; Diaz, M. & Saqui-Sannes, P. – Towards a formal specification of multimedia synchronization scenarios. *Annals of telecommunications* 49(5-6): 297-314, May 1994.
- [Sen96] Sénac, P. – *Contribution à la modélisation des systèmes multimédias et hypermédias*. Thèse, Toulouse - France, LAAS, June 1996.
- [Spi92] Spivey, J. M. – *The Z notation - A reference manual*. Second edition. EUA, Prentice Hall international series in computer science, Hoare, C.A.R. - editor, 1992.
- [Sto89] Stotts, P. D. & Furuta, R. – Petri-net-based hypertext: document structure with browsing semantics. *ACM Transactions on Information Systems* 7(1): 3-29, January 1989.
- [Sto90] Stotts, P. D. & Furuta, R. – Temporal hyperprogramming. *Journal of visual languages and computing* (1): 237-253, 1990.
- [Tom89] Tompa, F. W. – A data model for flexible hypertext database systems. *ACM Transactions on Information Systems* 7(1): 85-100, January 1989.
- [Wal83] Walter, B. – Timed Petri nets for modeling and analyzing protocols with time. *In Proceedings of the IFIP conference on protocol specification, testing and verification III*, North-Holland, Rudin, H. & West, C. - editors, 1983.
- [Wan95] Wang, H. K. & Wu, J. C. – Interactive hypermedia applications: a model and its implementation. *Software - Practice and Experience* 25(9): 1045-1063, September 1995.
- [Will96] Willrich, R.; Sénac, P.; Saqui-Sannes, P. & Diaz, M. – Towards hypermedia documents design. *Anais do XIV Simpósio Brasileiro de Redes de Computadores*, Fortaleza - Brasil: 473-491, 1996.
- [Woo94] Woo, M.; Qazi N. & Ghafoor, A. – A synchronization framework for communication of pre-orchestrated multimedia information. *IEEE Network*, January - February 1994.

Apêndice A

Este apêndice apresenta o modelo de objetos de uma ferramenta de apoio ao modelo XHMBS. A especificação é dada em termos do método Fusion de desenvolvimento orientado a objetos [Co94] e tem por base os modelos de objetos da ferramenta HySCharts e dos statecharts do ambiente Statsim, este último recuperado por engenharia reversa através do método FusionRE [Pen96].

Os principais elementos tratados por esse modelo são os *hypercharts*, os hiperdocumentos e os sistemas de apresentação e comunicação. Essa modelagem não considera objetos de interface relativos à autoria, seja do *hyperchart* ou do próprio hiperdocumento, como símbolos gráficos ou dispositivos de controle da interação entre a ferramenta e o especificador. O foco da modelagem apresentada é a estrutura dos elementos da especificação de um hiperdocumento em XHMBS, além das saídas determinadas pela sua apresentação.

Os *hypercharts* são modelados como uma extensão do modelo de objetos de statecharts do ambiente Statsim, possuindo como elementos diferenciais uma especialização adicional da classe *Transição* (*TransM:Nsync*) e uma especialização da classe *Evento Atômico* (*Evento Temporal*), representando, respectivamente, as *transições M:N sincronizadas* e os *eventos temporais*. Além disso, a classe básica *Estado* recebeu os atributos *LSC* ("Local Step Counter"), que deverá conter, para um estado *s*, o valor atualizado de *LSC* (*s*), de acordo com a definição apresentada no capítulo 4, e o atributo *T_hist* que, analogamente, deverá representar o valor atualizado de *T_hist* (*s*). Exceto a classe *TransDefault*, todas as demais especializações da classe *Transição* receberam o atributo *Histtemp*, cujo domínio deverá ser constituído de três valores constantes possíveis, representando, respectivamente, a não-associação da transição a qualquer tipo de história temporal, a associação da transição a *história temporal* e associação da transição a *história temporal recursiva*. Com isso um modelo de operações a ser criado pode determinar as demais restrições sobre a sintaxe dos *hypercharts*.

No modelo que especifica a autoria de um hiperdocumento, são apresentados os principais relacionamentos entre o autor de um hiperdocumento e os objetos que podem ser manipulados. Os principais relacionamentos são a edição dos objetos de um *hyperchart* subjacente e as associações entre esses objetos e as páginas e âncoras do hiperdocumento. Uma decisão não tomada neste trabalho é o suporte à autoria em ponto pequeno, ou seja, pode-se permitir, adicionalmente, que o autor crie páginas e âncoras e não simplesmente as manipule como objetos já criados (como arquivos, por exemplo).

O modelo de objetos que especifica a navegação de um hiperdocumento é o mais complexo entre os três apresentados neste apêndice. Os mapeamentos do modelo XHMBS são representados diretamente por relacionamentos, levando à semântica desejada para cada um deles. O mecanismo de apresentação da informação é definido pela classe *Sistema de Apresentação*, cuja principal característica é a utilização dos *pchannels* como elementos de interface. As classes *Menu Iniciar* e *Índice* especificam os objetos que possibilitam, respectivamente, o início da navegação de um hiperdocumento e a navegação através de um

índice. Note-se que esses objetos estão relacionados, assim como páginas e âncoras, a *pchannels*. Para esses objetos, especificamente, foi definida a sub-classe *Menu_Pchannel*, cuja implementação deve suportar a exibição de menus “cliqueáveis” através dos quais eventos específicos são gerados ou operações são invocadas.

A classe *Gatilho* especifica o objeto responsável por coletar os eventos gerados pelos *pchannels*, os quais devem corresponder a eventos definidos nos *hypercharts*, e disparar a simulação do *hyperchart*, a qual resultará na criação de um novo objeto *configuração*, com o qual pode-se apresentar as novas páginas e âncoras através dos respectivos *pchannels* na *área de display*.

Além disso, como suporte às aplicações distribuídas, se houver relacionamento entre um objeto *pchannel* e um objeto *cchannel*, este último será acionado e entregará os dados recebidos através da rede ao *pchannel* em questão. Os *cchannels* constituem o *sistema de apresentação* da ferramenta.

Adicionalmente, um objeto hiperdocumento possui o atributo *L* que, como em HMBS, define o nível de visibilidade desse hiperdocumento. As operações *show_hview*, *up* e *down* podem ser definidas no modelo de operações.

Esses relacionamentos dão a noção que norteia a extensão da ferramenta HySCharts para o suporte traduzido ao XHMBS.

Apêndice B

Este apêndice apresenta a expansão informal da especificação parametrizada do estado *MM_Apresent*, apresentado nas figuras 5.10 a 5.16, no capítulo 5. Para efeito de clareza do diagrama, as ações das transições associadas ao evento *inv* são especificadas somente por identificadores, cujos conteúdos podem ser vistos na tabela abaixo. Finalmente, após a apresentação da especificação expandida, apresenta-se um resumo de como os requisitos da apresentação foram satisfeitos.

a ₁	$LSC(A_iR) := 10000 - LSC(A_iF); d := 1^*$
a ₂	$LSC(A_iF) := 10000 - LSC(A_iR); d := 0^*$
a ₃	$LSC(V_iR) := 10000 - LSC(V_iF); d := 1^*$
a ₄	$LSC(V_iF) := 10000 - LSC(V_iR); d := 0^*$
a ₅	$LSC(I_iR) := 10000 - LSC(I_iF); d := 1$
a ₆	$LSC(I_iF) := 10000 - LSC(I_iR); d := 0$
a ₇	$LSC(I_2R) := 10000 - LSC(I_2F); d := 1$
a ₈	$LSC(I_2F) := 10000 - LSC(I_2R); d := 0$
a ₉	$LSC(I_2_NullR) := 10000 - LSC(I_2_NullF); d := 1$
a ₁₀	$LSC(I_2_NullF) := 10000 - LSC(I_2_NullR); d := 0$
a ₁₁	$LSC(T_1R) := 10000 - LSC(T_1F); d := 1$
a ₁₂	$LSC(T_1F) := 10000 - LSC(T_1R); d := 0$
a ₁₃	$LSC(T_2R) := 20000 - LSC(T_2F); d := 1$
a ₁₄	$LSC(T_2F) := 20000 - LSC(T_2R); d := 0$
a ₁₅	<i>imgpch.release_ch()</i>
a ₁₆	<i>imgpch.allocate_ch()</i>

* O índice *i* é devido à parametrização de estados. Assim, na verdade, as ações a₁, a₂, a₃ e a₄ representam, cada qual, um conjunto de 90 ações, cada uma correspondendo ao valor do índice *i* instanciado pela parametrização do estado AV_i, apresentada na figura 5.16.

Satisfação dos requisitos da apresentação multimídia pela modelagem em *hypercharts*

Requisito	Satisfação
1. Cada SV tem duração de 5 segundos.	Cada estado $V_i F$ ou $V_i R$, juntamente com uma transição associada ao evento temporal [258, 333, 408], especifica uma porção de cada SV. Cada seqüência de 15 estados $V_i F$ ($V_i R$) especificam uma SV com 5 segundos de duração. De acordo com a especificação parametrizada do estado AV_i , existem 90 estados $V_i F$ e 90 estados $V_i R$. Assim, esses estados especificam, 15 a 15, as seis SVs em ambos os sentidos de apresentação.
2. Cada SA tem duração de 10 segundos.	Cada estado $A_i F$ ($A_i R$), juntamente com a respectiva transição temporal associada ao evento temporal [258, 333, 408], especifica uma fração de cada SA. São necessários 30 estados $A_i F$ ($A_i R$) para especificar uma SA em um dos sentidos de apresentação, determinando que existem 3 SAs.
3. Cada I tem duração de 10 segundos.	As duas imagens são especificadas pelos estados $I_1 F$, $I_1 R$, $I_2 F$ e $I_2 R$ e pelas transições que partem deles. Além disso os estados $I_2_Null F$ e $I_2_Null R$ são utilizados para o balanceamento temporal do cenário de sincronização, facilitando a especificação da interação de inversão da direção da apresentação.
4. O "jitter" para imagens é de 2 segundos.	As transições temporais que partem dos estados que especificam imagens estão associados a eventos que possuem a forma $[x-2000, x, x+2000]$.
5. O primeiro T tem duração de 10 segundos.	Especificado pela transição temporal com origem no estado $T_1 F$, rotulada por [8000, 10 ⁴ , 12000].
6. O segundo T tem duração de 20 segundos.	Especificado pelos eventos temporais associados, à transição e ao arco de origem com origem nos estados $T_2 F$ e $T_2 R$.
7. O "jitter" para texto é de 2 segundos.	Especificado da mesma forma como ocorre com imagens.
8. A primeira SV, a primeira I e o primeiro T começam a ser apresentados ao mesmo tempo.	Isso é garantido pela ativação por "default" de todos os estados iniciais de cada cadeia de dados quando da ativação do estado $MM_Apresent$.
9. Quando a primeira SV termina, a segunda começa.	O encadeamento dos estados $V_i F$ (ou $V_i R$) por transições temporais garante a continuidade da apresentação nos termos desse requisito.
10. A segunda SV e a primeira SA terminam ao mesmo tempo. Quando terminam, a	Devido ao uso de sincronização do tipo <i>weak-and</i> , garante-se que todas as frações das SVs e SAs acabam

Requisito	Satisfação
terceira SV e a segunda SA começam a ser apresentadas ao mesmo tempo.	suas apresentações ao mesmo tempo por sincronização. Devido à forma contínua com que são seqüenciadas, garante-se também que as próximas cadeias iniciarão suas apresentações imediatamente após o final da apresentação das anteriores.
11. Quando a terceira SV termina, a quarta começa.	idem.
12. A quarta SV e a segunda SA acabam ao mesmo tempo. Quando terminam, a quinta SV e a terceira SA começam ao mesmo tempo.	idem.
13. Quando a quinta SA termina, a sexta começa a ser apresentada.	idem.
14. As SA e SV precisam ser sincronizadas a cada terço de segundo. O "jitter" para SAs e SVs é de 75 milissegundos para cada ponto de sincronização.	O uso de transições M:N sincronizadas com arcos de origem rotulados pelo evento temporal [258, 333, 408] garante a satisfação desse requisito.
15. A primeira I e o primeiro T terminam ao mesmo tempo. Quando terminam, a segunda I e o segundo T começam a ser apresentados ao mesmo tempo.	De forma similar ao que ocorre com áudio e vídeo, o uso de transições M:N sincronizadas garante a satisfação desse requisito.
16. A segunda imagem termina na metade do tempo de apresentação do segundo texto.	Respeitadas as variações internas previstas pelos "jitters" das cadeias de dados, dados por [8000, 10 ⁴ , 12000] para a segunda imagem e [18000, 20000, 22000] para o segundo texto, esse requisito é satisfeito
17. Quando a segunda I termina, a área de "display" é liberada e fica assim até o final da apresentação.	Satisfeito pelas ações <i>release_ch ()</i> e <i>allocate_ch ()</i> associadas às transições que partem dos estados I ₂ _F e I ₂ _Null_R.
18. A sexta SV, a terceira SA e o segundo T terminam ao mesmo tempo. A sexta SV e o segundo T permanecem na região de display até que o usuário solicite o término da apresentação ou siga a ligação de volta ao menu.	A transição M:N sincronizada do tipo <i>strong-or</i> que tem destino no estado <i>MM_ocioso</i> garante a satisfação desse requisito.
19. Cada uma das cadeias de dados tem origem em um único local.	Esse requisito é satisfeito, visto que os <i>pchannels</i> que apresentam as cadeias de dados não precisam, necessariamente, ser mapeados para algum <i>cchannel</i> . Se esse mapeamento não existe, uma aplicação local fica caracterizada.
20. Todas as fontes de dados são pré-orquestradas.	Pode ser satisfeito através do atributo <i>tipo_fonte</i> dos <i>pchannels</i> .

Requisito	Satisfação
21. Iniciar	Modelado pelo evento <i>iniciar</i> .
22. Saltar para frente	Modelado pelos eventos <i>saltar</i> e <i>tm (saltar, n)</i> e as respectivas ações disparadas pelas transições a que esses eventos estão associados.
23. Saltar para trás	Modelado pelos eventos <i>inv</i> , <i>saltar</i> e <i>tm (saltar, n)</i> , além das ações associadas às respectivas transições.
24. Reiniciar	Modelado pelo evento <i>reiniciar</i> , considerando as duas direções de apresentação.
25. Terminar	Modelado pelo evento <i>terminar</i> .
26. Congelar	Modelado pelo evento <i>congelar</i> .
27. Descongelar	Modelado pelo evento <i>descongelar</i> , associado a uma transição com história temporal.
28. Inverter a direção de apresentação	Modelado pelo evento <i>inv</i> , além das ações associadas às respectivas transições.
29. Alterar a velocidade da apresentação	Modelado pelo evento <i>alterar_veloc</i> e pelas ações associadas à respectiva transição.