

Luís Filipe dos Santos Gomes

# **Redes de Petri Reactivas e Hierárquicas**

- Integração de formalismos no projecto de  
sistemas reactivos de tempo-real -

Dissertação apresentada para obtenção do Grau de  
Doutor em Engenharia Electrotécnica, Especialidade  
de Sistemas Digitais, pela Universidade Nova de  
Lisboa, Faculdade de Ciências e Tecnologia

Lisboa

1997



## Sumário

Nesta dissertação faz-se a apresentação de uma nova classe de Redes de Petri, as Redes de Petri Reactivas e Hierárquicas (RdP-RH). O objectivo principal da proposta é o de suportar o projecto integrado de sistemas reactivos de tempo-real, permitindo, para além do apoio às diversas fases do ciclo de desenvolvimento, nomeadamente especificação, validação, verificação e realização, integrar submodelos especificados através de diferentes formalismos. Como exemplos representativos de sistemas reactivos de tempo-real refiram-se os sistemas embebidos, os sistemas de automação e os circuitos digitais de aplicação específica. De entre os formalismos tidos como interessantes, refiram-se, os formalismos típicos de especificação dos sistemas a eventos discretos passíveis de uma representação gráfica, como as máquinas de estado, os *statecharts* e as redes de Petri, bem como alguns dos formalismos genericamente designados como de controlo inteligente, como sistemas de produção de regras, de regras com imprecisão e de regras difusas.

A classe das Redes de Petri Reactivas (RdP-R), utilizada como núcleo das RdP-RH, é caracterizada como tomando as Redes de Petri Coloridas como classe de referência, às quais se adicionam capacidades de modelação de características não-autónomas, intrínsecas aos sistemas que se pretendem modelar. Discutem-se alguns aspectos ligados à sua realização, nomeadamente os temas de construção do espaço de estados e da resolução automática de conflitos. A introdução de três mecanismos distintos de estruturação hierárquica, denominados por decomposição horizontal, vertical e mista, conduz à definição das RdP-RH. Nelas se utilizam três tipos de nós denominados por macronós, metanós e supernós, associados aos três mecanismos propostos. Os mecanismos de estruturação hierárquica do modelo são complementados com a representação vectorizada dos nós do grafo.

Discute-se a aplicação das RdP-RH na modelação de *statecharts* e de formalismos de controlo inteligente, com ênfase para os controladores difusos, onde a necessidade de integrar controlo e processamento de dados permite utilizar cabalmente as capacidades das RdP-RH. Em torno da análise de trabalhos realizados na área de aplicação de “edifícios inteligentes”, utilizada como referência para o trabalho desenvolvido, identificam-se alguns temas em que se prevêem ou são desejados desenvolvimentos no futuro próximo recorrendo às RdP-RH.

## Abstract

This thesis presents a new class of Petri nets, named Hierarchical Reactive Petri Nets (HRPN). The main goal of this proposal is intended to support the integrated design of real-time reactive systems; it is two-fold: support the different phases of the development cycle, namely specification, validation, verification and implementation and integrate several sub-models specified through different formalisms. As relevant examples of real-time reactive systems, one may refer the following ones: embedded systems, automation systems and application specific digital circuits. Among the interesting formalisms, one may refer the discrete-event specification graphical formalisms, namely state machines, statecharts and Petri nets, as well the set of formalisms known for intelligent control, like production rule based, rules with imprecision and fuzzy rule systems.

The class of Reactive Petri nets (RPN), which are the core of HRPN, takes the Coloured Petri nets class as a reference and includes several non-autonomous modelling capabilities, associated with the target systems characteristics. Several implementation aspects will be discussed, namely the state space construction and automatic conflict resolution techniques. The addition of three mechanisms for hierarchical model organisation will generate the HRPN definition. They are called by horizontal, vertical and mixed decomposition schemes and use three specific types of nodes, namely macro-nodes, meta-nodes and super-nodes. The model structuring mechanisms are complemented with the node vector based representation capability.

Afterwards, several HRPN applications will be discussed, namely to the statechart modelling and to intelligent control formalisms, emphasising fuzzy controller modelling, where the need for simultaneous modelling of the control path and the data processing path will rely on unique HRPN capabilities. Around the analysis of several works in the area of intelligent buildings, which was used as a reference for the developed work, several subjects are identified, where the use of HRPN model applications are foreseen or interesting to consider.

## Sommaire

Dans cette thèse on présente une nouvelle classe de Réseaux de Petri, les Réseaux de Petri Réactifs et Hierarchiques (RdP-RH). L'objectif principal de cette proposition est celui de supporter le projet intégré de systèmes réactifs de temps-réel, permettant, au-delà de l'appui aux différentes phases du cycle de développement, notamment l'espécification, la validation, la vérification et l'implémentation, intégrer sous-modèles spécifiés à travers de différents formalismes. Comme exemples représentatifs des systèmes réactifs de temps-réel on doit référer le systèmes enfouie, les systèmes d'automatisme et les circuits digitales d'application spécifique. D'entre les formalismes considérés intéressants, il faut faire référence aux formalismes typiques de spécification des systèmes à événements discrets passibles d'une représentation graphique, comme les machines d'état, les *statecharts* et les réseaux de Petri, ainsi que quelques formalismes génériquement désignés comme de contrôle intelligent, comme les systèmes de production de règles, de règles imprécises et de règles diffuses.

La classe de Réseaux de Petri Réactifs (RdP-R) utilisée comme noyau des RdP-RH, est caractérisée comme ayant pris les Réseaux de Petri Colorés comme classe de référence, aux quels on ajoute des capacités de modelage des caractéristiques non-autonomes, intrinsèques aux systèmes que l'on veut modéliser. On discute quelques aspects liés à l'implémentation, notamment les thèmes de construction de l'espace des états et de la résolution automatique de conflits. L'introduction des trois mécanismes distincts de structuration hiérarchique, désignés par décomposition horizontal, vertical et mixte, conduit à la définition des RdP-RH, où l'on utilise trois types de noeuds désignés par macronoeuds, metanoeuds et supernoeuds, associés aux trois mécanismes proposés. Les mécanismes de structuration hiérarchique du modèle sont complétés avec la représentation vectorisée des noeuds du graphe.

On discute l'application des RdP-RH dans la modelage de *statecharts* et de formalismes de contrôle intelligent, avec un intérêt spécial pour les contrôleurs diffus, où la nécessité d'intégrer le contrôle et le calcul permet d'utiliser totalement les capacités des RdP-RH. Autour de l'analyse des travaux réalisés dans le domaine d'application des "immeubles intelligents", utilisée comme référence pour le travail développé, on identifie quelques thèmes où l'on prévoit ou désire des développements, dans un futur prochain, en faisant recours aux RdP-RH.



## Simbologia e Notações

### ***Geral***

|           |  |
|-----------|--|
| =         | igual a  |
| ≠         | diferente de   |
| >         | maior do que   |
| ≥         | maior ou igual a   |
| <         | menor do que   |
| ≤         | menor ou igual a   |
| /         | negação lógica   |
| +         | ou lógico  |
| *         | e lógico   |
| ∅         | conjunto vazio   |
| ∑         | somatório  |
| $A[i]$    | elemento $i$ do vector $A$   |
| $A[i..j]$ | vector $A$ com $(j-i+1)$ elementos, referidos através de $A[i] \dots A[j]$ , respectivamente |

### ***Redes de Petri***

|                                 |   |
|---------------------------------|---|
| $p$                             | lugar   |
| $t$                             | transição   |
| $\langle a, \dots, b \rangle$   | marca colorida com os atributos $a, \dots, b$       |
| $x \langle a, \dots, b \rangle$ | evento $x$ transportando os atributos $a, \dots, b$ |

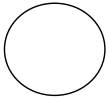

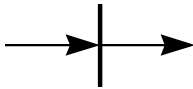
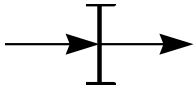
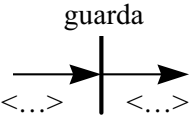
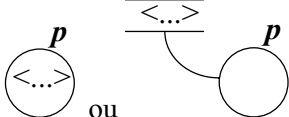
$m_0$           marcação inicial da rede de Petri

$m_i$           marcação da rede de Petri

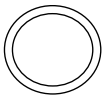

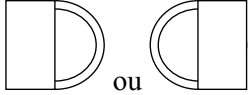
$m(p)$         marcação do lugar  $p$

## Notação gráfica



### Geral

|  |  |  |  |
|--|--|--|--|
| lugar  |   | arco                                       |   |
| transição assíncrona<br>(e arcos associados) |   | transição síncrona<br>(e arcos associados) |   |
| guarda (de transição)                        |  | marcação (de lugar)                        |  |

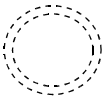

### Macronós

|            |   |                |   |
|------------|---|----------------|---|
| macrolugar |  | macrotransição |  |
| macrobloco |  |                |   |

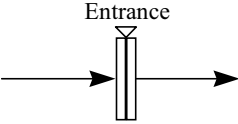
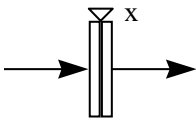
### Metanós

|           |   |               |   |
|-----------|---|---------------|---|
| metalugar |  | metatransição |  |
|-----------|---|---------------|---|

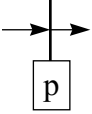
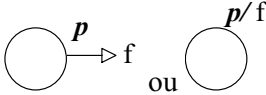
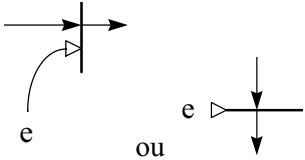
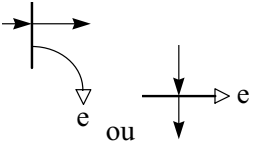
### Supernós

|            |   |                |   |
|------------|---|----------------|---|
| superlugar |  | supertransição |  |
|------------|---|----------------|---|

## Transições especiais de interface

|  |   |
|--|---|
| <p>transição síncrona<br/>de entrada</p>  | <p>transição síncrona<br/>de saída (em que <math>x</math> pode ser Stop, Reset ou Exit)</p>  |
|--|---|

## Características não-autónomas

|   |   |
|---|---|
| <p>prioridade (de transição)</p>                                     | <p>acção de saída <math>f</math><br/>(associada a lugar <math>p</math>)</p>  |
| <p>evento de entrada <math>e</math><br/>(associado a transição)</p>  | <p>evento de saída <math>e</math><br/>(associado a transição)</p>            |



# Índice de Matérias

|  |             |
|--|-------------|
| <b>Sumário .....</b>                                       | <b>iii</b>  |
| <b>Abstract.....</b>                                       | <b>iv</b>   |
| <b>Sommaire.....</b>                                       | <b>v</b>    |
| <b>Simbologia e Notações .....</b>                         | <b>vii</b>  |
| <i>Geral .....</i>   | <i>vii</i>  |
| <i>Redes de Petri .....</i>                                | <i>vii</i>  |
| <i>Notação gráfica .....</i>                               | <i>viii</i> |
| <b>Índice de Matérias.....</b>                             | <b>xi</b>   |
| <b>Índice de Figuras.....</b>                              | <b>xvii</b> |
| <b>Índice de Tabelas.....</b>                              | <b>xxii</b> |
| <b>Prefácio .....</b>                                      | <b>xxiv</b> |
| <i>Sobre a estrutura da tese .....</i>                     | <i>xxv</i>  |
| <i>Agradecimentos .....</i>                                | <i>xxv</i>  |
| <b>1. Sobre os objectivos.....</b>                         | <b>1</b>    |
| <i>1.1 Que sistemas a modelar.....</i>                     | <i>3</i>    |
| <i>1.2 Que objectivos para o presente trabalho.....</i>    | <i>6</i>    |
| 1.2.1 Integração de formalismos.....                       | 8           |
| 1.2.2 Aspectos de metodologia.....                         | 10          |
| <i>1.3 Que estratégia a utilizar na implementação.....</i> | <i>12</i>   |
| <i>1.4 Que atitude na investigação desenvolvida .....</i>  | <i>13</i>   |
| <i>1.5 A estrutura da tese .....</i>                       | <i>16</i>   |

---

|   |           |
|---|-----------|
| <b>2. As Redes de Petri Reactivas .....</b>                           | <b>19</b> |
| 2.1 <i>Formalismos para sistemas reactivos de tempo-real .....</i>    | 21        |
| 2.1.1 Máquinas de estados finitos .....                               | 23        |
| 2.1.2 Statecharts .....   | 26        |
| 2.1.3 Esterel .....   | 28        |
| 2.1.4 Redes de Petri (RdP) .....                                      | 29        |
| 2.1.4.a RdP Autónomas .....   | 31        |
| 2.1.4.b RdP Não-Autónomas .....                                       | 32        |
| 2.1.4.c Representação de arcos .....                                  | 34        |
| 2.2 <i>Descrição da classe proposta: as RdP-Reactivas .....</i>       | 36        |
| 2.2.1 Parte autónoma do modelo .....                                  | 37        |
| 2.2.2 Caracterização da arquitectura de execução .....                | 38        |
| 2.2.3 Modelação de entradas .....                                     | 43        |
| 2.2.4 Sincronismo em relação a eventos externos .....                 | 46        |
| 2.2.5 Utilização de prioridades .....                                 | 46        |
| 2.2.6 Evolução do estado do modelo .....                              | 47        |
| 2.2.7 Introduzindo dependências temporais .....                       | 49        |
| 2.2.8 Modelação de fluxos de controlo e de dados .....                | 52        |
| 2.2.9 Modelação de saídas .....                                       | 54        |
| 2.2.10 Condições e eventos internos .....                             | 55        |
| 2.2.11 Interdependência entre as partes autónoma e não-autónoma ..... | 57        |
| 2.2.12 Sobre a notação gráfica utilizada .....                        | 59        |
| 2.3 <i>Análise de um exemplo .....</i>                                | 61        |
| <b>3. Aspectos de verificação e de implementação .....</b>            | <b>67</b> |
| 3.1 <i>Sobre as técnicas de verificação e execução de RdP-R .....</i> | 69        |
| 3.1.1 Jogador-de-marcas versus jogador-de-estados .....               | 69        |
| 3.1.2 Compilação versus interpretação .....                           | 73        |
| 3.1.3 Da especificação à implementação .....                          | 74        |
| 3.1.3.a Métodos de análise aplicáveis .....                           | 75        |
| 3.1.3.b Estratégia para a geração de código .....                     | 76        |
| 3.2 <i>Sobre ambientes de desenvolvimento .....</i>                   | 77        |
| 3.3 <i>Sobre a construção e utilização do grafo de estados .....</i>  | 80        |
| 3.3.1 Das RdP-R a máquinas de estados finitos .....                   | 81        |
| 3.3.2 Execução em plataformas de baixo custo .....                    | 83        |

|           |  |            |
|-----------|--|------------|
| 3.3.3     | Obtenção de grafos de estados condensados .....  | 87         |
| 3.4       | <i>Resolução de conflitos</i> .....  | 88         |
| 3.4.1     | Classificação das soluções propostas .....   | 89         |
| 3.4.1.a   | Resolução não-equitativa.....  | 91         |
| 3.4.1.a.1 | Utilização de prioridades associadas às transições .....                                       | 91         |
| 3.4.1.a.2 | Construindo um espaço de estados não-equitativo.....   | 93         |
| 3.4.1.b   | Resolução determinística e equitativa .....  | 93         |
| 3.4.1.b.1 | O caso das RdP de escolha-livre .....  | 94         |
| 3.4.1.b.2 | Adicionando árbitros determinísticos e equitativos .....                                       | 95         |
| 3.4.1.b.3 | Implementações baseadas no espaço de estados .....   | 100        |
| 3.4.1.c   | Resolução não-determinística e equitativa .....  | 102        |
| 3.4.2     | Sistematização das soluções propostas.....   | 104        |
| 3.4.2.a   | Carta de selecção de solução .....   | 104        |
| 3.4.2.b   | Sobre a indicação do método de resolução de conflitos.....                                     | 106        |
| <b>4.</b> | <b>As Redes de Petri Reactivas e Hierárquicas.....</b>   | <b>107</b> |
| 4.1       | <i>Introdução</i> .....  | 109        |
| 4.1.1     | Relações com os modelos de RdP .....   | 111        |
| 4.2       | <i>Representações hierárquicas e modulares com RdP</i> .....                                   | 113        |
| 4.2.1     | Utilizando RdP de baixo-nível.....   | 114        |
| 4.2.2     | Utilizando RdP de alto-nível .....   | 120        |
| 4.2.2.a   | Utilizando RdP Coloridas .....   | 121        |
| 4.2.2.a.1 | Substituição de transições .....   | 122        |
| 4.2.2.a.2 | Substituição de lugares .....  | 124        |
| 4.2.2.a.3 | Evocação de transições.....  | 124        |
| 4.2.2.a.4 | Fusão de lugares e fusão de transições.....  | 125        |
| 4.2.2.a.5 | Canais de comunicação síncrona.....  | 126        |
| 4.2.2.b   | Utilizando RdP Predicado-Transição .....   | 127        |
| 4.2.2.c   | Utilizando representações orientadas por objectos .....  | 129        |
| 4.2.3     | Utilizando RdP não-autónomas .....   | 133        |
| 4.2.4     | Ênfases e lacunas dos trabalhos apresentados .....   | 133        |
| 4.3       | <i>Propostas para a modelação hierárquica e modular de sistemas reactivos de tempo-real</i> .. | 136        |
| 4.3.1     | Decomposição horizontal .....  | 137        |
| 4.3.1.a   | Utilizando macrolugares.....   | 140        |
| 4.3.1.b   | Utilizando macrotransições .....   | 141        |
| 4.3.1.c   | Utilizando macroblocos .....   | 143        |

---

|   |     |
|---|-----|
| 4.3.1.d Sobre as macro-redes .....                                      | 145 |
| 4.3.1.e Explorando a utilização de macronós .....                       | 147 |
| 4.3.2 Representação nodal estruturada através de vectores .....         | 150 |
| 4.3.2.a Semântica associada.....  | 151 |
| 4.3.2.b Tradução em RdP Coloridas.....                                  | 157 |
| 4.3.3 Decomposição vertical.....  | 158 |
| 4.3.3.a Utilizando metalugares .....                                    | 160 |
| 4.3.3.b Utilizando metatransições.....                                  | 163 |
| 4.3.3.c Sobre as meta-redes .....                                       | 164 |
| 4.3.3.c.1 Inicialização de uma meta-rede simples .....                  | 165 |
| 4.3.3.c.2 Desactivação de uma meta-rede simples .....                   | 167 |
| 4.3.3.c.3 Nós da meta-rede .....  | 168 |
| 4.3.3.c.4 Comunicação ascendente.....                                   | 169 |
| 4.3.3.c.5 Execução de modelos contendo metanós.....                     | 171 |
| 4.3.3.c.6 Conceito de história .....                                    | 174 |
| 4.3.3.c.7 Utilizando meta-redes com história .....                      | 176 |
| 4.3.3.c.8 Reinicializando meta-redes com história.....                  | 178 |
| 4.3.3.d Utilizando meta-redes solidárias.....                           | 180 |
| 4.3.3.e Análise breve de aplicações .....                               | 181 |
| 4.3.3.e.1 Aplicação à modelação de sistemas de tempo-real.....          | 181 |
| 4.3.3.e.2 Modelação de subprogramas .....                               | 183 |
| 4.3.3.e.3 Modelação de statecharts através de RdP-RH.....               | 185 |
| 4.3.4 Decomposição mista .....  | 188 |
| 4.3.4.a Sobre as super-redes .....                                      | 192 |
| 4.3.4.b Procedimentos de inserção das subpáginas .....                  | 193 |
| 4.3.4.b.1 Modelando os nós internos às sub-redes .....                  | 193 |
| 4.3.4.b.2 Modelando a activação das sub-redes .....                     | 194 |
| 4.3.4.b.3 Modelando a desactivação das sub-redes .....                  | 196 |
| 4.3.4.b.4 Métodos de obtenção do modelo plano .....                     | 198 |
| 4.3.4.c Retorno à modelação de statecharts através de RdP-R.....        | 199 |
| 4.4 Exemplo de aplicação .....  | 202 |
| 4.4.1 Utilização de macronós.....                                       | 204 |
| 4.4.2 Utilização de representação estruturada baseada em vectores ..... | 206 |
| 4.4.3 Utilização de metanós .....                                       | 207 |
| 4.4.4 Utilização de supernós .....                                      | 209 |

|  |            |
|--|------------|
| <b>5. Modelação de técnicas de controlo inteligente .....</b>                                | <b>213</b> |
| 5.1 <i>Introdução</i> .....  | 215        |
| 5.2 <i>Modelação de sistemas de regras</i> .....   | 215        |
| 5.2.1 <i>Sistemas de regras de produção</i> .....  | 215        |
| 5.2.2 <i>Regras não-catóricas</i> .....  | 217        |
| 5.2.3 <i>Sistemas de regras de produção difusas</i> .....                                    | 218        |
| 5.3 <i>Modelação de controlo difuso</i> .....  | 225        |
| 5.3.1 <i>Modelando variáveis linguísticas difusas</i> .....                                  | 225        |
| 5.3.2 <i>Estrutura de um controlador difuso</i> .....  | 225        |
| 5.3.3 <i>Codificação difusa</i> .....  | 226        |
| 5.3.4 <i>Inferência difusa</i> .....   | 227        |
| 5.3.5 <i>Utilização de qualificadores</i> .....  | 229        |
| 5.3.6 <i>Descodificação difusa</i> .....   | 231        |
| 5.3.7 <i>Representação alternativa de controladores difusos</i> .....                        | 231        |
| 5.3.8 <i>Sobre a execução</i> .....  | 232        |
| 5.3.9 <i>Organização hierárquica de controladores</i> .....                                  | 235        |
| 5.4 <i>Aplicação a controladores programáveis</i> .....                                      | 237        |
| 5.5 <i>Retomando a resolução automática de conflitos</i> .....                               | 245        |
| <b>6. Edifícios inteligentes: um conceito potencialmente mobilizador.....</b>                | <b>247</b> |
| 6.1 <i>Introdução</i> .....  | 249        |
| 6.2 <i>Um conceito pontencialmente mobilizador para a indústria e para os serviços</i> ..... | 255        |
| 6.3 <i>Enquadramento das propostas a apresentar</i> .....                                    | 258        |
| 6.4 <i>A necessidade de novos ocupantes: os domots</i> .....                                 | 261        |
| 6.5 <i>Perspectivas de modelação</i> .....   | 264        |
| 6.6 <i>Ambientes de desenvolvimento e exploração em edifícios</i> .....                      | 266        |
| 6.7 <i>Arquitecturas de referência</i> .....   | 269        |
| 6.7.1 <i>Interligação de sistemas e dispositivos</i> .....                                   | 269        |
| 6.7.1.a <i>Redes e protocolos comuns</i> .....   | 269        |
| 6.7.1.b <i>Arquitectura desenvolvida</i> .....   | 272        |
| 6.7.2 <i>Interligação lógica</i> .....   | 275        |
| 6.7.3 <i>Caracterização dos serviços disponíveis</i> .....                                   | 278        |
| 6.7.3.a <i>Serviços comuns</i> .....   | 279        |

|   |            |
|---|------------|
| 6.7.3.b Serviços específicos do servidor de imagens.....            | 280        |
| 6.7.3.c Serviços específicos ao servidor de sensores.....           | 281        |
| 6.7.4 Exploração: o exemplo do Campus-Guard .....                   | 282        |
| 6.8 <i>Análise de casos</i> .....                                   | 284        |
| 6.8.1 Modelação de estruturas sensoriais e de visualização .....    | 284        |
| 6.8.2 Modelação de sistemas distribuídos .....                      | 287        |
| <b>7. Conclusões.....</b>   | <b>291</b> |
| 7.1 <i>Sobre o trabalho realizado</i> .....                         | 293        |
| 7.1.1 Enquadramento nacional.....                                   | 293        |
| 7.1.2 Publicação de resultados.....                                 | 294        |
| 7.1.3 Participação em projectos e transferência de resultados ..... | 294        |
| 7.2 <i>Sobre os resultados obtidos</i> .....                        | 295        |
| 7.3 <i>Sobre o trabalho futuro</i> .....                            | 297        |
| <b>8. Bibliografia .....</b>  | <b>299</b> |
| <b>Epílogo.....</b>   | <b>319</b> |

## Índice de Figuras

|  |    |
|--|----|
| Figura 1-1 - O papel de ponte entre formalismos, desempenhado pelas RdP-R e RdP-RH. ....   | 10 |
| Figura 1-2 - As funções de integração de formalismos e linguagem intermédia desempenhadas pelas RdP-R e RdP-RH. ....   | 11 |
| Figura 1-3 - Execução de RdP.....  | 13 |
| Figura 1-4 - Desenvolvimento de metodologia com RdP Coloridas.....   | 13 |
| Figura 1-5 - Os três aspectos principais na resolução de um problema [Calvez, 93]. ....  | 14 |
| Figura 1-6 - Ciclo de desenvolvimento da metodologia proposta. ....  | 15 |
| Figura 2-1 - Representação de máquinas de estados finitos: a) diagrama de estados; b) fluxograma. ...  | 24 |
| Figura 2-2 - Exemplo de statechart (a) e (b) espaço de estados associado. ....   | 28 |
| Figura 2-3 - Especificação de um autómato: (a) programa em Esterel; (b) statechart equivalente. ....   | 29 |
| Figura 2-4 - (a) RdP equivalente às especificações da Figura 2-3; (b) Disparo de uma transição. ....   | 30 |
| Figura 2-5 - Estrutura de referência do controlador.....   | 39 |
| Figura 2-6 - Tipos de sinais, sua representação interna e estruturas de dados associadas. ....   | 43 |
| Figura 2-7 - Compactação da representação do fluxo de controlo. ....   | 45 |
| Figura 2-8 - Representação compacta e modelo RdP equivalente. ....   | 45 |
| Figura 2-9 - Execução recorrendo a várias definições temporais, indicando-se os tempos de execução das várias tarefas, individualmente e por grupos. ....                                  | 49 |
| Figura 2-10 - Utilização do atributo de duração “ttl”. ....  | 51 |
| Figura 2-11 - RdP-R elementar e espaço de estados associado. ....  | 53 |
| Figura 2-12 - Seleção de eventos através de atributos de marca.....  | 58 |
| Figura 2-13 - Geração de marcas considerando variáveis externas associadas a eventos. ....   | 59 |
| Figura 2-14 - Notação gráfica proposta: exemplos. ....   | 59 |
| Figura 2-15 - Diagrama de estados modelando a comunicação de um cliente com um servidor. ....  | 61 |
| Figura 2-16 - Statechart modelando a comunicação de um cliente com dois servidores. ....   | 62 |
| Figura 2-17 - Statechart equivalente ao apresentado na Figura 2-16.....  | 63 |
| Figura 2-18 - RdP modelando a comunicação de um cliente e dois servidores.....   | 64 |
| Figura 2-19 - RdP-R equivalente à RdP da Figura 2-18. ....   | 64 |
| Figura 3-1 - Mapa de soluções para a execução das RdP-R. ....  | 69 |
| Figura 3-2 - A solução à medida: a) no plano “dimensão do modelo a executar versus recursos computacionais disponíveis”; b) adaptação das RdP-R a plataformas de execução específicas. ... | 71 |

|  |     |
|--|-----|
| Figura 3-3 - Etapas do desenvolvimento.....  | 74  |
| Figura 3-4 - Protótipo de ambiente de desenvolvimento [Gomes & al, 94b].....   | 78  |
| Figura 3-5 - Ecrã de edição típico (em [Gomes & al, 94b]).....   | 79  |
| Figura 3-6 - Especificação para simulação e resultado (em [Gomes & al, 94b]).....  | 80  |
| Figura 3-7 - (a) uma RdP-R simples; (b) estados de marcação possíveis; (c) grafo de estados associado. ....  | 81  |
| Figura 3-8 - Um diagrama de estados.....   | 85  |
| Figura 3-9 - Diagrama de contactos associado ao grafo de estados da Figura 3-8. ....   | 86  |
| Figura 3-10 - Especificação Palasm associada à máquina de estados da Figura 3-8. ....  | 87  |
| Figura 3-11 - Representação de uma taxonomia de soluções para a resolução automática de conflitos.91   |     |
| Figura 3-12 - Resolução de conflitos em RdP de escolha-livre: a) modelo inicial; b) modelo comportamentalmente equivalente sem conflitos; c) e d) introdução de diferentes índices de equidade. ....         | 95  |
| Figura 3-13 - Resolução de um conflito em problema de exclusão mútua: a) modelo inicial; b) modelo quasi-equivalente sem conflito; c) modelo quasi-equivalente sem conflito e inicialização equitativa. .... | 96  |
| Figura 3-14 - Resolução determinística e equitativa do problema de exclusão mútua: representação hierárquica. ....   | 97  |
| Figura 3-15 - Outra resolução de conflito: a) modelo inicial; b) modelo não-equivalente sem conflitos; c) modelo quasi-equivalente sem conflitos. ....   | 98  |
| Figura 3-16 - Resolução de conflito: a) modelo inicial; b) espaço de estados associado incluindo não-determinismo; c) árbitro auxiliar; d) espaço de estados alterado.....                                   | 101 |
| Figura 3-17 - Estrutura modular de blocos, aplicável na resolução de conflitos em implementações baseadas na execução do espaço de estados. ....   | 102 |
| Figura 3-18 - Resolução de conflito: a) modelo inicial; b) espaço de estados associado incluindo não-determinismo; c) árbitro auxiliar; d) espaço de estados alterado.....                                   | 103 |
| Figura 3-19 - Carta de soluções para a resolução de conflitos.....   | 105 |
| Figura 4-1 - Modelação de um evento não-primitivo.....   | 114 |
| Figura 4-2 - Modelação hierárquica utilizando refinamentos sucessivos. ....  | 114 |
| Figura 4-3 - Esquema básico para modelação de subprogramas.....  | 116 |
| Figura 4-4 - Um modelo SA (N) com as suas três máquinas de estado elementares (N1, N2 e N3) modelando uma exclusão mútua em relação a um recurso comum. ....   | 117 |
| Figura 4-5 - p2 é um lugar implícito; está “em paralelo” com p3 e p4.....  | 119 |
| Figura 4-6 - (a) p1 e p3 são lugares implícitos; (b) p2 é lugar implícito. ....  | 119 |
| Figura 4-7 - Exemplo de uma rede HHPN contendo outras duas sub-redes HHPN. ....  | 121 |
| Figura 4-8 - Mecanismo de substituição de transição: (a) superpágina; (b) subpágina; (c) modelo plano equivalente.....   | 123 |

|   |     |
|---|-----|
| Figura 4-9 - Utilização de canais síncronos e RdP equivalente. ....   | 126 |
| Figura 4-10 - Uma RdP Predicado Transição Hierárquica (a) e Rdp PrT equivalente (b). ....   | 128 |
| Figura 4-11 - Os seis padrões básicos de suporte à abstracção. ....   | 128 |
| Figura 4-12 - Vistas externa e interna de um superlugar. ....   | 130 |
| Figura 4-13 - Notação gráfica ambígua: a) especificação; b) interpretação como fusão de lugares; c) interpretação como substituição de lugares. ....  | 135 |
| Figura 4-14 - Notação gráfica de macronós: a) macrolugar, b) macrotransição, c) e d) macrobloco. ....   | 139 |
| Figura 4-15 - Modelação de uma sequência de controlo através de um macrolugar: a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente. ....  | 140 |
| Figura 4-16 - Modelo recorrendo a uma macrotransição: a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente. ....   | 142 |
| Figura 4-17 - Utilização de um macrobloco: a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente. ....  | 144 |
| Figura 4-18 - Modelação de um sistema de produção em linha. ....  | 147 |
| Figura 4-19 - Modelação modular do sistema de produção em linha. ....   | 148 |
| Figura 4-20 - Modelo de baixo-nível do sistema de produção em linha. ....   | 149 |
| Figura 4-21 - Modelo utilizando blocos do sistema de produção em linha. ....  | 149 |
| Figura 4-22 - Modelo utilizando uma estrutura de blocos do sistema de produção em linha. ....   | 150 |
| Figura 4-23 - Semântica associada à região IV da Tabela 4-1. ....   | 154 |
| Figura 4-24 - Semântica associada à região VI da Tabela 4-1. ....   | 155 |
| Figura 4-25 - Semântica associada à região II da Tabela 4-1. ....   | 155 |
| Figura 4-26 - Utilização de características estruturadas através de vectores em modelos comportamentalmente equivalentes: a) representação não permitida; b) representação estruturada válida; c) representação colorida válida. .... | 158 |
| Figura 4-27 - Metalugar: representação gráfica e modelo associado na superpágina. ....  | 160 |
| Figura 4-28 - Modelo hierárquico parcial de controlo de aquecimento: a) nível de topo; b) nível 2; c) visão planar da representação hierárquica. ....   | 161 |
| Figura 4-29 - Utilização de metalugares em: a) lugares monomarcados; b) lugares plurimarcados. ....   | 162 |
| Figura 4-30 - Metatransição: a) representação na superpágina; b) modelo equivalente. ....   | 163 |
| Figura 4-31 - Activação de uma meta-rede: a) transição de entrada; b) utilização com metalugar; c) utilização com metatransição. ....   | 166 |
| Figura 4-32 - Lugares de saída: a) identificação na meta-rede; b) identificação alternativa; c) superpágina. ....   | 170 |
| Figura 4-33 - Execução paralela de modelos com metanós. ....  | 172 |
| Figura 4-34 - Modelo executável associado à meta-rede <i>Temp</i> da Figura 4-28(b). ....   | 174 |
| Figura 4-35 - Utilização de metalugares plurimarcados com história: a) modelo especificado; b) modelo equivalente. ....   | 176 |

|   |     |
|---|-----|
| Figura 4-36 - Modelo executável associado à meta-rede <i>Temp</i> da Figura 4-28(b) quando referida com o atributo de história.....   | 178 |
| Figura 4-37 - Modelo executável associado à meta-rede <i>Temp</i> da Figura 4-28(b) quando utilizada com o atributo de história e capacidade de reinicialização.....                    | 179 |
| Figura 4-38 - Modelação de interrupções forçada e possível.....   | 183 |
| Figura 4-39 - Modelação de chamada de sub-rotina. ....  | 184 |
| Figura 4-40 - Exemplo de modelos comportamentalmente equivalentes: a) statechart; b) páginas de RdP-RH. ....  | 187 |
| Figura 4-41 - Statechart da Figura 4-40(a): a) sua representação textual; b) espaço de estados associado. ....  | 189 |
| Figura 4-42 - Representação de supernós: a) superlugar; b) supertransição.....  | 191 |
| Figura 4-43 - Modelos de execução das páginas da Figura 4-40(b).....  | 200 |
| Figura 4-44 - Modelo plano comportamentalmente equivalente ao statechart da Figura 4-40(a). ....  | 200 |
| Figura 4-45 - Submodelo relativo à RdP da Figura 4-44. ....   | 201 |
| Figura 4-46 - Máquina de estados associada à RdP-R da Figura 4-44.....  | 202 |
| Figura 4-47 - Máquina de estados resultante da apresentada na Figura 4-46. ....   | 202 |
| Figura 4-48 - Sistema produtor-consumidor: a) armazém com capacidade infinita; b) armazém com capacidade igual a dois.....  | 203 |
| Figura 4-49 - Sistema produtor-consumidor: representação hierárquica com dois níveis. ....  | 203 |
| Figura 4-50 - Descrições das páginas da RdP da Figura 4-49.....   | 205 |
| Figura 4-51 - Máquina de estados associada à RdP da Figura 4-48(b) e da Figura 4-49.....  | 205 |
| Figura 4-52 - Sistema com um produtor e dois consumidores com um armazém com capacidade igual a dois: a) modelo plano; b) recurso a macronós; c) recurso a macronós com estrutura. .... | 206 |
| Figura 4-53 - Acomodar a introdução de um novo consumidor.....  | 207 |
| Figura 4-54 - Modelo hierárquico multinível de um sistema produtor - dois consumidores. ....  | 208 |
| Figura 4-55 - Execução paralela do modelo hierárquico multinível da Figura 4-54.....  | 209 |
| Figura 4-56 - RdP-R plana comportamentalmente equivalente à RdP-RH da Figura 4-54. ....   | 210 |
| Figura 4-57 - Representação do modelo da Figura 4-56. ....  | 212 |
| Figura 5-1 - A RdP associada ao Exemplo 5.1.....  | 217 |
| Figura 5-2 - Grafo de estados parcial associado ao modelo da Figura 5-1. ....   | 217 |
| Figura 5-3 - Modelação de excepções em RdP-R. ....  | 217 |
| Figura 5-4 - A RdP-R associada ao Exemplo 5.2.....  | 222 |
| Figura 5-5 - Grafo de estados parcial associado à Figura 5-4.....   | 222 |
| Figura 5-6 - Modelo alternativo equivalente para o Exemplo 5.2.....   | 223 |
| Figura 5-7 - Execução do modelo RdP-R descompactado. ....   | 223 |
| Figura 5-8 - A RdP-R associada ao Exemplo 5.3.....  | 224 |

|   |     |
|---|-----|
| Figura 5-9 - Modelação da fase de codificação difusa: a) modelo RdP-R; b) conjuntos difusos associados à variável de entrada <i>Com</i> ; c) marcação gerada na situação de <i>Com=25</i> ..... | 226 |
| Figura 5-10 - Representação linguística de regra difusa.....  | 227 |
| Figura 5-11 - Representação tabular de um controlador difuso.....   | 228 |
| Figura 5-12 - Modelo RdP-R associado ao Exemplo 5.4.....  | 229 |
| Figura 5-13 - Modelação de qualificadores: a) num passo de computação; b) em dois passos.....   | 230 |
| Figura 5-14 - Alternativas comuns para implementação em hardware (adaptado de [Costa et al., 95]).  | 233 |
| Figura 5-15 - Modelo comportamentalmente equivalente ao da Figura 5-12.....   | 235 |
| Figura 5-16 - Junção de controladores hierarquicamente organizados.....   | 237 |
| Figura 5-17 - Especificação inicial de controlo do sistema de ar condicionado.....  | 238 |
| Figura 5-18 - Partição difusa das variáveis Temperatura (a) e Ventoíinha (b).....   | 239 |
| Figura 5-19 - Código ABL para fase de codificação difusa.....   | 241 |
| Figura 5-20 - Código ABL para fase de inferência difusa.....  | 241 |
| Figura 5-21 - Código ABL para fase de descodificação difusa.....  | 242 |
| Figura 5-22 - Modelo RdP-R de regras difusas associadas a nível de topo.....  | 244 |
| Figura 6-1 - A pirâmide do Edifício Inteligente (adaptado de [DEGW & Teknibank, 95]).....   | 252 |
| Figura 6-2 - Dependência do custo de uma funcionalidade do nível de integração de sistemas.....   | 260 |
| Figura 6-3 - Mapeamento de actividades.....   | 261 |
| Figura 6-4 - Características principais de um domot.....  | 263 |
| Figura 6-5 - Agentes solicitadores de recursos em edifícios.....  | 263 |
| Figura 6-6 - Diferentes perspectivas para a modelação do edifício.....  | 264 |
| Figura 6-7 - Modelo de dados parcial dos detectores de presença.....  | 265 |
| Figura 6-8 - Actividades de desenvolvimento decorrentes do conceito de edifício inteligente.....  | 267 |
| Figura 6-9 - Caracterização do ambiente de desenvolvimento.....   | 268 |
| Figura 6-10 - Arquitectura lógica de referência para interligação física de dispositivos e sistemas.....  | 272 |
| Figura 6-11 - Arquitectura lógica de referência para interligação de processos em estações de controlo.....   | 276 |
| Figura 6-12 - Configuração laboratorial de teste.....   | 278 |
| Figura 6-13 - Estruturação do sinóptico de monitoração de instalações.....  | 282 |
| Figura 6-14 - Interface típico do Campus-Guard.....   | 283 |
| Figura 6-15 - Modelo RdP-R simplificado de uma estrutura de alarme.....   | 285 |
| Figura 6-16 - Interacção entre um emissor e um receptor.....  | 288 |
| Figura 6-17 - Decomposição em objectos: (a) interacção, (b) emissor e (c) receptor.....   | 289 |

## Índice de Tabelas

|  |     |
|--|-----|
| Tabela 2-1 - Tipos de arcos e semântica associada.....   | 35  |
| Tabela 3-1 - Análise da dinâmica do diagrama de estados da Figura 3-8.....   | 85  |
| Tabela 3-2 - Expressões de activação e desactivação. ....  | 85  |
| Tabela 4-1 - Caracterização das situações a considerar, em termos da dependência em relação à representação estruturada através de vectores..... | 153 |
| Tabela 5-1 - Computação dos factores de confiança das conclusões.....  | 219 |
| Tabela 5-2 - Tipos de regras, RdP-R associadas e factores de confiança de conclusão associados..   | 220 |
| Tabela 5-3 - Agregação de factores de confiança. ....  | 221 |
| Tabela 6-1 - Classificação das funções do edifício inteligente (segundo [McKinley, 88]). ....  | 251 |
| Tabela 6-2 - Tipos de solução característicos da inteligência predial (segundo [DEGW & Teknibank, 95]). ....                                     | 254 |
| Tabela 6-3 - Classificação dos processos em execução. ....   | 277 |



## Prefácio

Esta dissertação aborda algumas temáticas pouco comuns no panorama técnico-científico português. É essa, claramente, a situação em relação às Redes de Petri, onde, a avaliar pelo número de comunicações e demais publicações técnico-científicas de autores portugueses, o tema tem uma divulgação quase insignificante. É essa também a situação nas abordagens à área de aplicação aos edifícios inteligentes.

Ao ter assumido, conjuntamente com o orientador científico desta tese, Prof. Adolfo Steiger Garção, a decisão de tentar promover trabalho inovador nestas temáticas, desde cedo ficou claro que a inexistência de uma “consciência colectiva” com história, necessitaria de ser compensada por diversas formas.

A primeira, aparentemente a mais dolorosa, traduz-se na necessidade de fornecer (praticamente) toda a força-motriz para fazer avançar o trabalho. De alguma forma, a sensação com que se chega ao final desta fase do trabalho é a de ter utilizado uma motorizada a pedais de reduzida potência, tal o esforço dispendido. Para além de ter sido necessário seleccionar um caminho predominantemente de esforço, isto é, de subida, foi necessário garantir o andamento de algumas actividades adjacentes. Só o elevado número de apoios recebidos, que tentarei identificar posteriormente, me permitiram fortalecer e prosseguir.

A segunda, porventura a mais fácil e agradável, foi a de procurar acompanhar os eventos da comunidade científica de Redes de Petri e de através da interacção resultante ir encontrando um trilho “seguro”. Foi possível constatar que “o caminho se faz caminhando”, como dizia o poeta. É-me grato verificar que, como uma das consequência dessa interacção, será possível trazer em 1998 a principal conferência sobre Redes de Petri a Portugal (19<sup>th</sup> International Conference on Applications and Theory of Petri Nets) e, desta forma, tentar contribuir para inverter a situação actual de alheamento da comunidade científica nacional em relação às Redes de Petri.

Por diversas vezes ao longo deste percurso, as Redes de Petri apareceram como potencialmente integradoras de atitudes de modelação e resultados de diferentes comunidades científicas. Essa é uma das razões para a extensão dos modelos formais inicialmente propostos pelo Prof. Carl Adam Petri a imensas áreas de aplicação, para o que foram sendo propostas novas classes de Redes de Petri. Esse é

também o papel da classe de Redes de Petri proposta neste trabalho e que colheu em mim o mais forte empenhamento.

## ***Sobre a estrutura da Tese***

Optou-se por uma estruturação do texto da tese em capítulos (sete, no total). As partes principais de cada capítulo foram denominadas por secções, enquanto que as partes resultantes da decomposição das secções foram referidas por subsecções (independentemente do grau de subdivisão verificado). De uma forma geral, pretendeu-se que cada capítulo pudesse ser lido quase que separadamente.

As obras referidas no final do capítulo sobre bibliografia, nomeadamente o “Grande Dicionário da Língua Portuguesa”, de José Pedro Machado [Machado J.P., 90] e o “Prontuário Ortográfico Moderno” de Manuela Parreira e J. Manuel de Castro Pinto [Parreira & Pinto, 85], foram utilizadas para esclarecer as dúvidas de escrita surgidas durante a preparação da tese. No entanto, algumas palavras, como *implementação*, *software*, *hardware* e *firmware*, entre outras, foram utilizadas nesta tese, tendo em conta o seu estado de aceitação na linguagem comum.

Os provérbios populares que abrem alguns dos capítulos (e que me permitiram unir valores morais e de afectividade), foram, na sua maioria, extraídos da obra de José Pedro Machado, “O Grande Livro dos Provérbios” [Machado J.P., 96].

## ***Agradecimentos***

Inúmeras pessoas contribuíram, voluntária ou involuntariamente, para a realização desta tese. Impossível será, certamente, referi-las de modo exaustivo. No entanto, desejo deixar-lhes expressos os meus sinceros agradecimentos. Em particular, desejo agradecer aos que estiveram mais directamente envolvidos.

Em primeiro lugar desejo expressar a minha profunda gratidão ao Professor Doutor Adolfo Steiger Garção que, para além de orientador científico dos trabalhos desenvolvidos, soube encontrar os equilíbrios entre o “orientar” e o “criar espaço para aprender”, necessariamente presentes em todos os processos de formação, e com particular relevância em trabalhos de investigação como o presente. Os meus agradecimentos por tudo quanto me apoiou e incentivou quer na participação em projectos, quer na elaboração de propostas, quer nos meios de suporte que permitiu encontrar, bem como nas tarefas que me foram confiadas de coordenação de estágios e de equipas de projecto.

Em segundo lugar agradeço a colaboração de um elevado número de colegas, estagiários e alunos finalistas de Engenharia que tive a oportunidade de coordenar. A Anikó Costa, Carlos Soares, João

Paulo Barros e Pedro Próspero Luís, como representantes especiais desse grupo e colaboradores responsáveis durante a realização de vários protótipos, desejo expressar a minha gratidão.

A João Paulo Barros o meu obrigado pelas imensas conversas (também) sobre Redes de Petri, bem como pela leitura atenta de versões preliminares de alguns capítulos desta tese.

Aos Docentes, Investigadores, Técnicos e Funcionários com quem agradavelmente interagi do Departamento de Engenharia Electrotécnica da FCT, do Centro de Robótica Inteligente do UNINOVA, nas várias metamorfoses da sua existência, e do Departamento de Informática da FCT, quero expressar os meus sinceros agradecimentos pelo ambiente criado. De modo particular, a João Paulo Pimentão, colega docente voluntarioso, pelo companheirismo no combate a “muitos incêndios”, a Ricardo (*Kadu*) Rabelo, pela energia positiva que conosco partilhou, a Duarte Guerreiro, para além da disponibilidade para apoio técnico permanente, pela amizade demonstrada, e a Isabel Gomes, pela paciência, quero expressar os meus agradecimentos.

A Manuel Silva e Kurt Jensen, como representantes da comunidade de Redes de Petri com quem mais interagi, o meu reconhecimento pelo exemplo do modo de estar nessas actividades.

A José Pedro Machado, meu professor de Português no Ensino Secundário, quer pelos ensinamentos que ainda hoje em mim perduram (tolere-me os que, entretanto, esqueci), quer como representante (involuntário) do elevado número de excelentes Professores que tive a ventura de ter tido, desde a Primária à Universidade e que contribuíram para a formação da minha consciência cívica e profissional, desejo expressar a minha gratidão.

Os meus agradecimentos às Instituições que suportaram os trabalhos associados e onde estes foram desenvolvidos, nomeadamente ao Departamento de Engenharia Electrotécnica (DEE) da FCT-UNL e ao Centro de Robótica Inteligente (CRI) do UNINOVA, bem como às Instituições que, pontualmente, suportaram algumas das actividades, nomeadamente à NATO, pela participação numa Escola de Verão, ao programa ESPRIT, pelo suporte dado pelo projecto 21017-INNOVA na fase final dos trabalhos e ao programa CYTED, pela participação em *workshops* e reuniões de projecto, bem como em conferências internacionais. É-me grato verificar que, tendo sido autor da primeira Candidatura a Doutoramento apresentada ao DEE da FCT-UNL, é esta também a primeira tese de Doutoramento completamente desenvolvida dentro da sua existência.

Para finalizar, a minha comovida gratidão para minha Mãe Donzília e memória de meu Pai José, pela dedicação e esforço que a minha formação representou para os seus recursos de então, e à Ana pelo seu constante apoio e encorajamento, e por tudo o mais.

*Luis Gomes*

*Julho 1997*



# 1. Sobre os objectivos

*Diz-me com quem andas, dir-te-ei quem és.  
(Provérbio popular)*

## *Sumário do capítulo*

Os objectivos principais do presente capítulo são o de identificar os objectivos desta tese e de caracterizar os sistemas para os quais se pretende modelar o comportamento. Pretende-se, desta forma, deixar expressos quais os domínios técnico-científicos que mais contribuíram para o trabalho desenvolvido e quais as áreas de aplicação que mais o condicionaram.

A caracterização do trabalho da tese, no espírito do provérbio referido no topo desta página, será, assim, o resultado a obter neste capítulo.

Este capítulo é composto pelas seguintes secções:

- 1.1 Que sistemas a modelar*
- 1.2 Que objectivos para o presente trabalho*
- 1.3 Que estratégia a utilizar na implementação*
- 1.4 Que atitude na investigação desenvolvida*
- 1.5 A estrutura da tese*

Na primeira secção caracterizam-se os sistemas reactivos de tempo-real, identificados como sistemas alvo de modelação.

Na segunda secção apresentam-se os objectivos que foram considerados para os trabalhos de que esta tese é um resultado. Esboça-se a identificação de algumas contribuições inovadoras resultantes dos trabalhos realizados.

Os métodos de execução dos modelos a propôr são sumariamente esboçados na terceira secção (serão detalhados em capítulo posterior).

Na quarta secção tornar-se-á explícita a atitude de engenharia subjacente aos trabalhos desenvolvidos, baseada numa caracterização em três vertentes fundamentais que se beneficiam e condicionam mutuamente: teoria, ferramentas e aplicações.

Finalmente, a estrutura da presente tese é apresentada na quinta secção.

# 1. Sobre os objectivos

## 1.1 *Que sistemas a modelar*

Não é recente a afirmação de que “a literatura em engenharia de software, linguagens de programação e projecto de sistemas e de hardware está repleta de comunicações que descrevem métodos para especificar e projectar sistemas grandes e complexos” [Harel & Pnueli, 85: pp. 477]. Os autores da afirmação, utilizaram-na para se questionarem sobre a razão que os terá levado a propôr um novo método de especificação.

Importa iniciar o presente trabalho por alguma forma de balanço ou reflexão sobre a mesma afirmação. Em [Harel & Pnueli, 85], primeiro trabalho, de que temos conhecimento, onde se introduzem os statecharts<sup>1</sup> (a caracterizar em capítulo posterior), a resposta encontrada identificava vários tipos de sistemas computacionais e de processamento de dados em relação aos quais existiam diferentes tipos de consensos em termos das atitudes básicas para o projecto. Aí, em alternativa às várias dicotomias mais ou menos tradicionais para classificação de sistemas, dependendo das comunidades científicas em causa, emergiu uma caracterização inovadora baseada na distinção entre sistemas transformacionais e sistemas reactivos.

Na primeira categoria, a dos sistemas transformacionais, estão incluídos os sistemas que são descritos como operando num esquema “clássico”, isto é, recebendo dados de entradas, realizando-lhe transformações e fornecendo resultados ou saídas. Em [Berry & Gonthier, 92] esta categoria veio a ser subdividida, identificando-se os sistemas transformacionais propriamente ditos e os sistemas interactivos. Nos primeiros, os dados de entrada são fornecidos no início do processamento e os resultados obtidos no final. Nos segundos, é possível obter dados intermédios e fornecer mais dados de entrada durante o processamento; neste caso, o sistema interage continuamente com o ambiente,

---

<sup>1</sup> Statecharts: termo para o qual se desconhece tradução e se prefere não utilizar nenhuma; propostas de tradução são “estadogramas” (por analogia com “fluxograma” versus “flow charts”) ou “diagramas de Harel”, resultante da tradução de “Harel diagrams” encontrado nalguma literatura de divulgação de produtos comerciais.

com uma velocidade de reacção determinada pelo sistema (como exemplos, refiram-se os sistemas operativos ou os interfaces com o utilizador).

Na categoria dos sistemas reactivos, incluem-se todos aqueles em que é necessária a resposta contínua (reacção) aos estímulos do mundo exterior e onde, em geral, não se realizam computações ou transformações de dados, e se garante o acompanhamento da evolução do mundo exterior. Deste modo, estes sistemas não se acomodam facilmente a descrições em termos de funções ou transformações.

Tomando a caracterização apresentada, não são difíceis de encontrar exemplos de sistemas reactivos. De entre os que fazem parte da experiência quotidiana do autor, refiram-se os sistemas de automação industrial e predial, bem como os sistemas mecatrónicos (incluindo os sistemas digitais específicos).

As principais características dos sistemas reactivos são as seguintes, segundo [Halbwachs, 93]:

- envolvem concorrência de actividades; normalmente, uma forma conveniente de caracterizar este tipo de sistemas utiliza a identificação de componentes paralelas e cooperantes;
- obedecem a requisitos temporais rigorosos, envolvendo quer o ritmo associado à variação das entradas, quer os tempos de resposta das entradas para as saídas;
- são determinísticos, isto é, as saídas do sistema são unicamente determinadas pelas entradas e pelos instantes da sua ocorrência (isto é, dependendo do estado interno actual); esta é uma característica também perseguida neste trabalho (de notar que em [Harel & Pnueli, 85] esta característica não era considerada intrínseca aos sistemas reactivos); o determinismo presente na especificação deve ser preservado na sua implementação; esta característica permite uma distinção clara em relação aos sistemas interactivos, a maioria dos quais são, intrinsecamente, não determinísticos; como exemplos, refiram-se os interfaces de utilizador, onde o instante de atendimento a um estímulo exterior não pode, de modo geral, ser determinado de modo rigoroso;
- a segurança e a robustez são objectivos particularmente importantes; estando, normalmente, presentes em sistemas críticos, de que dependem vidas e bens, é necessária a utilização de métodos de projecto especialmente rigorosos, complementados com técnicas de verificação formal;

- normalmente, são realizados parte em *software*<sup>2</sup>, parte em *hardware*; por razões históricas, de custo ou de desempenho, parte dos sistemas reactivos são realizados exclusivamente em hardware; a maior parte são, no entanto, implementados em hardware e software; a separação das duas partes é realizada em fases avançadas do projecto, quando se recorre a técnicas de co-design<sup>3</sup>.

Uma taxonomia complementar, classifica os sistemas em termos dos tempos de reacção em relação às necessidades do ambiente onde se inserem. Desta forma, identificam-se os sistemas de tempo-real, como aqueles que satisfazem restrições dos seus tempos de resposta a estímulos externos [Laplante, 93]. De entre estes, refiram-se como de especial interesse os sistemas de tempo-real forte<sup>4</sup>, neste trabalho simplesmente referidos por sistemas de tempo-real, em que o não cumprimento de um tempo de resposta a um estímulo especificado conduz a uma falha do sistema. Complementarmente, embora de interesse reduzido para o presente trabalho, identificam-se os sistemas de tempo-real fraco<sup>5</sup>, em que o não cumprimento de um tempo de resposta conduz apenas a uma degradação do desempenho do sistema, mas não a uma falha, bem como os sistemas de tempo-real frouxo<sup>6</sup>, em que se tolera uma baixa probabilidade no não cumprimento de um tempo de resposta que conduza a uma falha.

Procedendo à sobreposição das duas taxonomias apresentadas (sistemas transformacionais, sistemas interactivos e sistemas reactivos versus sistemas de tempo-real e sistemas sem restrições no tempo de resposta), verifica-se que sistemas classificáveis em qualquer dos grupos da primeira taxonomia podem ser classificados como sistemas de tempo-real. Isto porque as duas taxonomias referidas são complementares: na primeira, o motor da classificação centra-se nos métodos de especificação, enquanto na segunda são as questões de implementação e operação que se constituem nos critérios de classificação.

Desta forma, um sistema interactivo de tempo-real pode ser descrito como satisfazendo as características de um sistema interactivo (no sentido atrás definido), que satisfaz restrições temporais (referidas comumente por condicionantes de tempo-real), isto é, garantia de reacção dentro de tempos pré-definidos. De notar que, embora satisfaça as restrições em termos dos tempos de resposta especificados, isto é, seja de comportamento previsível, não se exige uma evolução determinística, em

---

<sup>2</sup> Optou-se pela utilização dos termos “software”, “hardware” e “firmware”, sem tentativa de tradução, devido à sua ampla utilização.

<sup>3</sup> Co-design: termo para o qual se desconhece tradução e se prefere não propôr alguma; a interpretação do termo corresponde à de “projecto concorrente ou simultâneo”; no entanto, o termo “design” é, normalmente, traduzido ao longo do trabalho por “projecto”.

<sup>4</sup> “Tempo-real forte”, tradução de “hard real-time”.

<sup>5</sup> “Tempo-real fraco”, tradução de “soft real-time”.

<sup>6</sup> “Tempo-real frouxo”, tradução de “firm real-time”.

sentido estrito. Como exemplo, refira-se um sistema operativo de tempo-real e os mecanismos de comutação de processos e atendimento de interrupções disponíveis: de um modo geral, não se garante a ordem de atendimento de vários eventos externos, embora se garantam tempos de resposta máximos associados a cada um deles.

Os sistemas que se constituem em alvo principal para as propostas a apresentar neste trabalho são referidos por sistemas reactivos complexos de tempo-real ou simplesmente por sistemas reactivos de tempo-real, identificando-se, deste modo, claramente as ênfases colocadas. Este tipo de sistemas, da forma como serão caracterizados, são frequentemente referidos na literatura como sistemas embebidos de tempo-real<sup>7</sup> (veja-se, por exemplo, [Calvez, 93]). Nestes sistemas importa considerar, para além das características puramente reactivas, capacidades de acomodar descrições em termos de funções ou transformações, ou seja, para além do tratamento do fluxo de controlo, importa considerar o fluxo de dados.

Um sistema reactivo complexo de tempo-real, ou simplesmente sistema reactivo de tempo-real, é caracterizado, tendo por base o proposto em [Berry & Gonthier, 92], como sendo constituído por três partes:

- um interface com o ambiente em que se insere, normalmente com características interactivas, responsável pela aquisição de entradas, actuação das saídas e incluindo gestão de interrupções;
- um ou mais núcleos reactivos, responsáveis pela computação das saídas, em termos da reacção a um conjunto de entradas;
- um nível de gestão de dados, realizando tarefas transformacionais, sobre o controlo dos núcleos reactivos.

## **1.2 Que objectivos para o presente trabalho**

Os objectivos deste trabalho centram-se na proposta de um formalismo particularmente vocacionado para os sistemas reactivos de tempo-real, com características adequadas para ser utilizado numa metodologia de projecto integrado. As três partes constituintes deste tipo de sistemas, como caracterizadas no final da secção anterior, devem ser contempladas. Por razões de legibilidade, entre outras, a adopção de uma notação gráfica para traduzir o modelo do sistema apresenta-se de elevado

---

<sup>7</sup> “Sistemas embebidos de tempo-real”, tradução de “real-time embedded systems”.

interesse (“*uma imagem vale mais que mil palavras*”). Desta forma, o formalismo a propôr contemplará os seguintes aspectos:

- ⇒ aplicável nos vários níveis característicos do sistema, nomeadamente na modelação do interface com o exterior, da evolução do estado do sistema e das tarefas de transformação de dados subjacentes; deste modo, pretende-se modelar de modo uniforme e integrado os fluxos de controlo e de dados associados ao sistema;
- ⇒ suporte a todas as fases do seu ciclo de vida do sistema, desde a especificação, passando pela verificação de propriedades e validação da solução, e incluindo a produção de documentos para a sua implementação, teste e manutenção;
- ⇒ suporte à integração de especificações inicialmente produzidas com formalismos distintos; como será enfatizado em parágrafos seguintes, a capacidade de integrar num único modelo, submodelos parciais e complementares, especificados de forma diversa, deverá ser uma característica fundamental de qualquer metodologia de projecto de sistemas;
- ⇒ flexibilizar as formas de implementação, permitindo utilizar, de modo uniforme, durante quase todas as fases do desenvolvimento, o mesmo conjunto de métodos e ferramentas; a distinção de procedimentos será realizada apenas nas fases associadas ao tipo de realização pretendido.

Foi, obviamente, um conjunto ambicioso de objectivos, que não é satisfeito por nenhum dos formalismos ou ambientes de desenvolvimento conhecidos. Na realidade, embora os objectivos indicados (quando considerados separadamente) sejam próximos de preocupações expressas nas propostas de outros formalismos e metodologias, em nenhum deles se consegue a integração num mesmo modelo dos aspectos associados aos fluxos de dados e de controlo, bem como a capacidade de utilização concomitante de modelos especificados através de formalismos distintos, permitindo-se, deste modo, a sua validação e verificação simultâneas. É um traço distintivo fundamental entre a atitude de princípio assumida no presente trabalho e a assumida nos restantes trabalhos conhecidos.

Este conjunto de objectivos foi cumprido apenas parcialmente, como se documentará no restante do presente trabalho e continuará a ser uma meta para futuros desenvolvimentos. Serviu, todavia, de referência em relação ao caminho que se pretendeu seguir e conduziu às propostas das RdP Reactivas e das RdP Reactivas e Hierárquicas.

Nas subsecções seguintes abordar-se-ão dois aspectos dos objectivos referidos.

### 1.2.1 Integração de formalismos

Como comentário prévio, importa deixar claro que quando ao longo deste trabalho se mencionar “sistema”, não se está a limitar a referência a sistemas particulares, realizados em software, em hardware ou ainda de forma mista. Todos eles são endereçados pelo modelo a propôr, sendo a decisão sobre o modo de realização (normalmente referida por implementação) independente da especificação. Como exemplos de sistemas comuns em que se pretende utilizar a metodologia subjacente a esta tese, refiram-se os sistemas embebidos, os sistemas de automação e os circuitos digitais de aplicação específica.

Deste modo, o tipo de sistemas que se pretende modelar necessitam não apenas de evoluir em função dos estímulos recebidos, através da sua parte reactiva, como também proceder à computação de dados, através da sua componente transformacional.

É muito diversificado o tipo e número de formalismos de especificação pretensamente candidatos adequados para a modelação destes tipos de sistemas.

Importa, assim, tendo como referência a metáfora da colher e do garfo, utilizada<sup>8</sup> pelo Prof. Lotfi Zadeh, deixar claro que, embora determinado formalismo seja utilizável para a modelação de um determinado sistema, a sua adequação pode ser questionável. Ou, de outra forma, embora a utilização de um determinado formalismo seja possível em relação a uma aplicação, torna-se mais interessante poder utilizar formalismos alternativos, considerando as questões da engenharia associada. Como caso típico do que se pretende ilustrar, refira-se o processo de controlo cuja descrição é demasiado complexa para permitir uma caracterização matemática formal minimamente rigorosa (em termos de equações diferenciais ou outras, como tradicionalmente realizado nas teorias de controlo clássico) e em que uma descrição linguística dos procedimentos de controlo é possível através da observação da operação do sistema. Nestes casos, a utilização de formalismos de controlo inteligente pode constituir-se em solução. Em particular, caracterizações em termos de controladores difusos, como propostos em [Mamdani, 74] ou em [Takagi & Sugeno, 83], tem vindo a ser largamente utilizadas para a resolução de problemas de controlo de difícil modelação matemática.

---

<sup>8</sup> Contida numa mensagem electrónica para a *fuzzy-mailing-list* ([fuzzy-mail@dbai.tuwien.ac.at](mailto:fuzzy-mail@dbai.tuwien.ac.at)), datada de 10 de Fevereiro de 1997, que se transcreve, em parte, seguidamente:

*“As you know, there are some who claim that anything that can be done with fuzzy logic and possibility theory can be done with probability theory. This claim is a special case of a more general claim of the form: “anything that can be done with method A can be done with method B”.*

*A good example of this dictum is the following.*

*Anything that can be eaten with a spoon, e.g., soup, can be eaten with a fork. True. The only problem is that you may run out” of “time and patience.”*

Identifica-se, desta forma, um aspecto central neste trabalho: o de permitir proceder à integração de especificações produzidas com formalismos distintos, viabilizando a selecção do formalismo mais adequado para a especificação de cada uma das partes do sistema.

De entre os formalismos de elevado interesse para serem considerados na especificação de sistemas reactivos de tempo-real, no sentido atrás definido, identificam-se vários grupos:

- os associados à modelação de sistemas a eventos discretos, em que se recorre a uma caracterização do sistema em termos de estados observáveis no sistema e das transições possíveis entre esses estados; de entre os formalismos deste grupo, referem-se as máquinas de estados finitos ou autómatos finitos, os statecharts e as Redes de Petri (referidas, neste texto por RdP);
- os associados à modelação baseada em técnicas clássicas de inteligência artificial, de que se realçam os sistemas de regras de produção e de raciocínio impreciso;
- os associados à modelação baseada em técnicas de inteligência computacional, onde se inclui o controlo difuso.

Obviamente que, nas especificações de interesse a considerar não se incluem as que colidem com as características atrás apresentadas para os sistemas alvo, reactivos de tempo-real, nomeadamente no que se refere ao tempo e recursos necessários para a sua execução. Este comentário é particularmente válido para algumas especificações utilizando os formalismos de Inteligência Artificial e Inteligência Computacional, em que a execução associada (computacionalmente muito exigente) necessita de sistemas complexos em termos dos recursos necessários e para os quais nem sempre é possível garantir computação em tempo previsível.

As RdP serão utilizadas como o formalismo com características adequadas para satisfazer os objectivos propostos, nomeadamente de garantir a integração dos formalismos referidos e de ser aplicável nas várias partes constituintes dos sistemas de tempo-real. Será proposta uma nova classe de RdP, denominada por RdP Reactivas<sup>9</sup> e referida abreviadamente por RdP-R. Com base nessa classe, as Redes de Petri Reactivas e Hierárquicas<sup>10</sup> serão, posteriormente, definidas e referidas abreviadamente por RdP-RH.

---

<sup>9</sup> “RdP Reactivas”, traduzível por “Reactive Petri nets”.

<sup>10</sup> “Redes de Petri Reactivas e Hierárquicas”, traduzível por “Hierarchical Reactive Petri nets”.

Na Figura 1-1 realça-se o papel de ponte entre os formalismos de interesse, a desempenhar pelas RdP-R e RdP-RH.

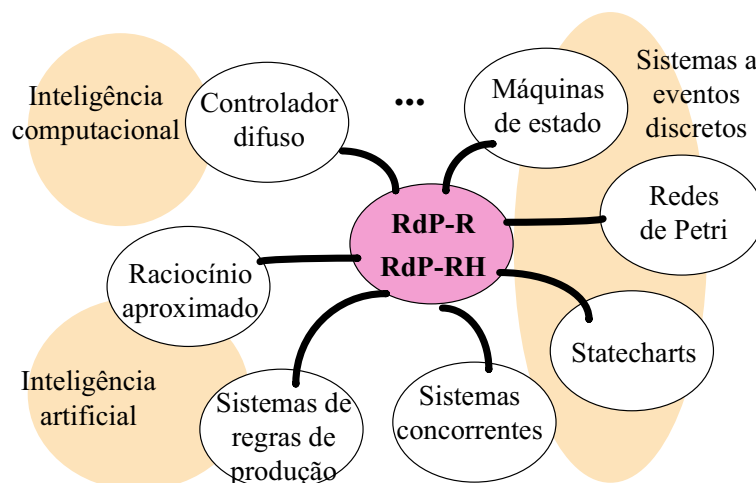


Figura 1-1 - O papel de ponte entre formalismos, desempenhado pelas RdP-R e RdP-RH.

### 1.2.2 Aspectos de metodologia

Um outro aspecto significativo no presente trabalho refere-se à metodologia a adoptar, nomeadamente em relação a questões de uniformidade na especificação e flexibilidade na implementação, consideradas em duas vertentes:

- i) a de suporte a abordagens sistémicas, em que o sistema é considerado globalmente, permitindo, através da identificação das várias partes constituintes, definir formas de implementação distintas para cada um dos componentes; incluem-se neste grupo as atitudes normalmente associadas às técnicas de co-design, em que as decisões do tipo de implementação a utilizar (hardware, software ou firmware) podem ser adiadas para as fases finais do projecto, dependendo de características específicas de desempenho e custo que se pretendam atingir;
- ii) a de suporte a diversas soluções de implementação, do ponto de vista das plataformas de execução e das linguagens utilizadas para a sua programação, incluindo a capacidade de produzir, de modo automático, código executável para diferentes plataformas.

As RdP-R/RdP-RH comportam-se como uma linguagem intermédia, neutra em relação ao grupo dos formalismos de especificação e ao grupo das linguagens de execução, permitindo a sua tradução mútua.

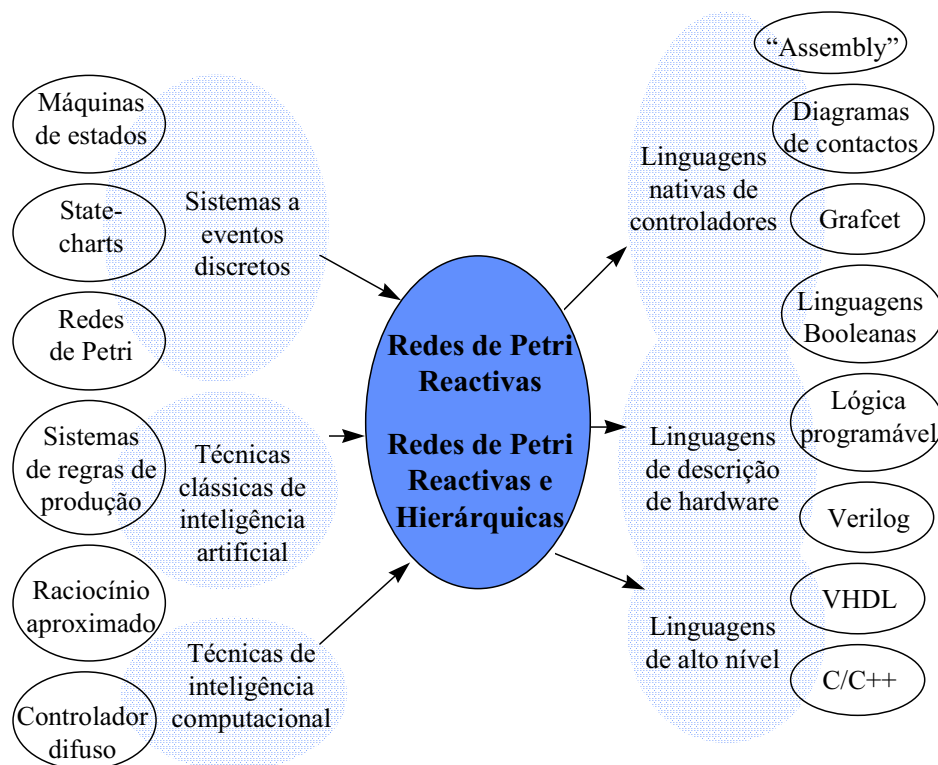


Figura 1-2 - As funções de integração de formalismos e linguagem intermédia desempenhadas pelas RdP-R e RdP-RH.

A Figura 1-2 evidencia as características de linguagem neutra em relação a diferentes formalismos de entrada, com capacidades de produzir especificações executáveis para um alargado leque de plataformas, onde se incluem, para além dos sistemas computacionais comuns, de reduzido custo ou de elevado desempenho, os sistemas embebidos de controlo ou automação, bem como os dispositivos de lógica programável.

Desta forma, identificam-se como interessantes, do ponto de vista da existência de tradutores específicos, vários grupos de linguagens de suporte à implementação:

- no primeiro grande grupo encontram-se as linguagens independentes da plataforma de implementação, vocacionadas para a modelação de sistemas de média e elevada complexidade; de referir as linguagens de alto nível normalmente utilizadas na programação de sistemas de tempo-real, nomeadamente C e C++, bem como as linguagens de alto nível vocacionadas para a descrição de hardware, de que o VHDL e o Verilog são, porventura, os exemplos mais significativos (referências a linguagens como Ada e a outras linguagens específicas para tempo-real seriam, igualmente, adequadas, embora não consideradas nos protótipos desenvolvidos e previstos);

- no segundo grande grupo encontram-se as linguagens de descrição de hardware que recorrem a arquitecturas configuráveis de referência para a sua implementação, como é o caso dos dispositivos de lógica programável e de que o PALASM [PALASM, 82] [PALASM, 90] é uma das mais remotas e significativas referências;
- no terceiro grande grupo encontram-se as linguagens específicas a determinadas soluções, nomeadamente as linguagens associadas aos controladores lógicos programáveis, utilizados normalmente em automação industrial e predial, de onde se notabilizam os diagramas de contactos<sup>11</sup>, as linguagens booleanas e textuais e o SFC (de que o Grafset [Grafset, 92] é o antecessor directo), integradas na norma IEC-1131 [IEC, 92].

### **1.3 Que estratégia a utilizar na implementação**

Pretendendo-se utilizar abordagens de decomposição funcional dos sistemas, de modo a possibilitar quer diferentes formalismos de especificação para cada uma das suas partes, quer diferentes formas de implementação, importa clarificar o modo como essas especificações são integradas e implementadas.

A metodologia proposta é a de tradução individual de cada especificação parcial num modelo RdP-R ou RdP-RH. Os vários modelos RdP-R/RdP-RH serão executados em paralelo e comunicarão entre si, quando necessário, quer para passagem de parâmetros, quer para sincronização, através de variáveis de entrada e saída.

Considerando a complexidade intrínseca a um modelo RdP-R ou RdP-RH e as características de algumas plataformas de execução previstas, são consideradas duas atitudes de referência para a execução das RdP-R/RdP-RH:

- i) a primeira resultante da execução directa do modelo, utilizando os tradicionais jogadores-de-marcas<sup>12</sup> (*token-player*, como normalmente referidos);
- ii) a segunda resultante da análise prévia da especificação e obtenção do seu espaço de estados associado.

A Figura 1-3 ilustra a metodologia seguida para obtenção de código executável a partir das especificações apresentadas.

---

<sup>11</sup> “Diagramas de contactos”, tradução de “ladder diagrams”.

<sup>12</sup> “Jogador-de-marcas”, tradução de “token-player”.

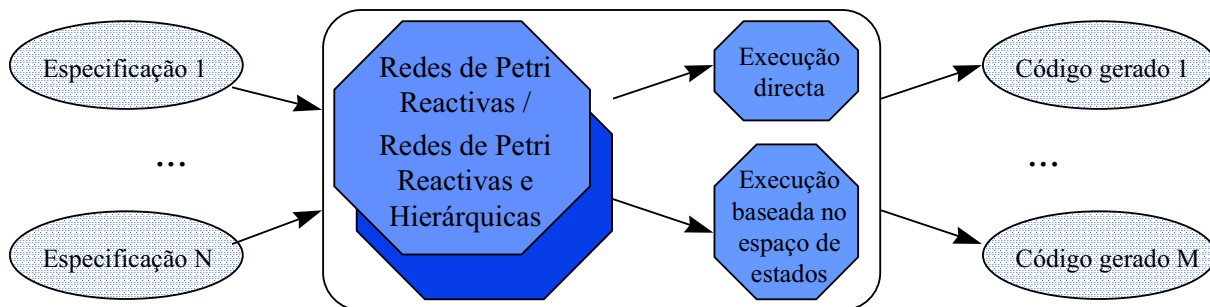


Figura 1-3 - Execução de RdP.

Em capítulo posterior as questões relativas à implementação serão retomadas com detalhe, nomeadamente a caracterização das arquitecturas para execução. No entanto, importa desde já referir que a construção do espaço de estados é encarada não apenas como um suporte à verificação das propriedades do modelo (como normalmente realizado), mas também oferecendo suporte à implementação, particularmente adequado quando as plataformas de execução dispõem de reduzidos recursos de computação (embora apenas aplicável em situações em que o modelo é de “reduzida” dimensão).

#### 1.4 Que atitude na investigação desenvolvida

A Figura 1-4 sintetiza a visão subjacente ao desenvolvimento das RdP Coloridas e seguida por Kurt Jensen [Jensen, 90] e Christensen [Christensen, sd].

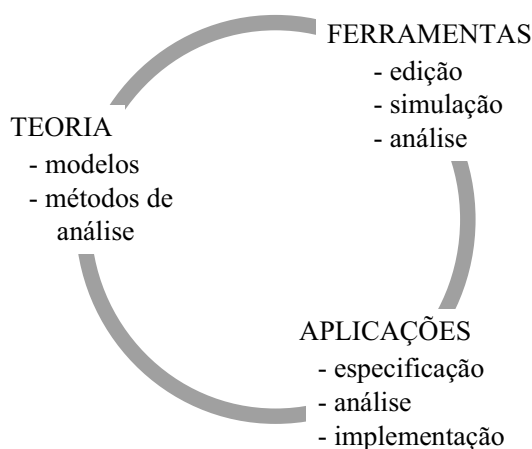


Figura 1-4 - Desenvolvimento de metodologia com RdP Coloridas.

Nela se evidencia a forte inter-relação entre teoria, ferramentas e aplicações, estando os três aspectos integrados no processo de desenvolvimento de uma metodologia e contribuindo sinergeticamente nos processos de desenvolvimento associados a cada um deles.

É uma visão marcadamente de Engenharia, que não descurando os aspectos formais associados à definição do modelo e dos métodos de análise aplicáveis, utilizou as aplicações como motor de desenvolvimento para ferramentas de suporte à aplicação do modelo. É uma fórmula plenamente bem sucedida, pelo menos do ponto de vista dos objectivos académicos, se forem considerados os inúmeros trabalhos desenvolvidos a partir das RDP Coloridas e das aplicações abordadas utilizando o ambiente Design/CPN, resultado de um esforço pesado de desenvolvimento por parte de um consórcio composto pela empresa de software e análise de sistemas MetaSoft e uma universidade (através do Prof. Kurt Jensen). Já do ponto de vista comercial, as conclusões não são tão auspiciosas. O ambiente Design/CPN, começando por ser um produto da MetaSoft, Inc. de custo elevado, é, actualmente, um produto mantido pela Universidade de Aarhus, utilizável sem encargos financeiros.

Uma caracterização tripartida, como a proposta em [Christensen, sd] e [Jensen, 90], é comumente seguida nas metodologias de projecto de sistemas. Citando outra abordagem, como exemplo, em [Calvez, 93], na sua “Metodologia de Concepção de Sistemas Electrónicos”, uma caracterização ligeiramente diferente é seguida, identificam-se três aspectos básicos para a resolução de um problema: os métodos, as ferramentas e as técnicas (ver Figura 1-5). As técnicas dão o suporte directo à implementação, enquanto as ferramentas são utilizadas para as implementar e os métodos permitem evoluir da definição do problema até à sua implementação.

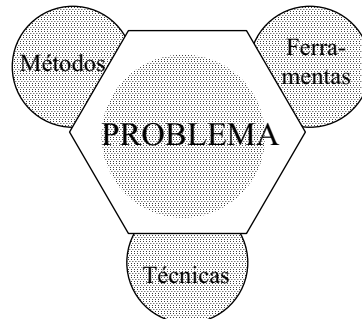


Figura 1-5 - Os três aspectos principais na resolução de um problema [Calvez, 93].

Também nos trabalhos que esta tese materializa, os três aspectos identificados por Jensen e Christensen evoluíram simultaneamente, condicionando-se mutuamente. Os desenvolvimentos nas três áreas identificadas beneficiaram das sinergias resultantes das actividades em cada uma das outras duas áreas. De referir, que algum do trabalho desenvolvido ou proposto foi unicamente possível considerando a articulação da participação do autor em projectos de investigação e desenvolvimento e em redes de cooperação científica e tecnológica, na orientação de projectos de fim de curso, bem como na participação na orientação de trabalhos de mestrado.

A Figura 1-6 sintetiza as principais referências do presente trabalho em termos da trilogia apresentada.

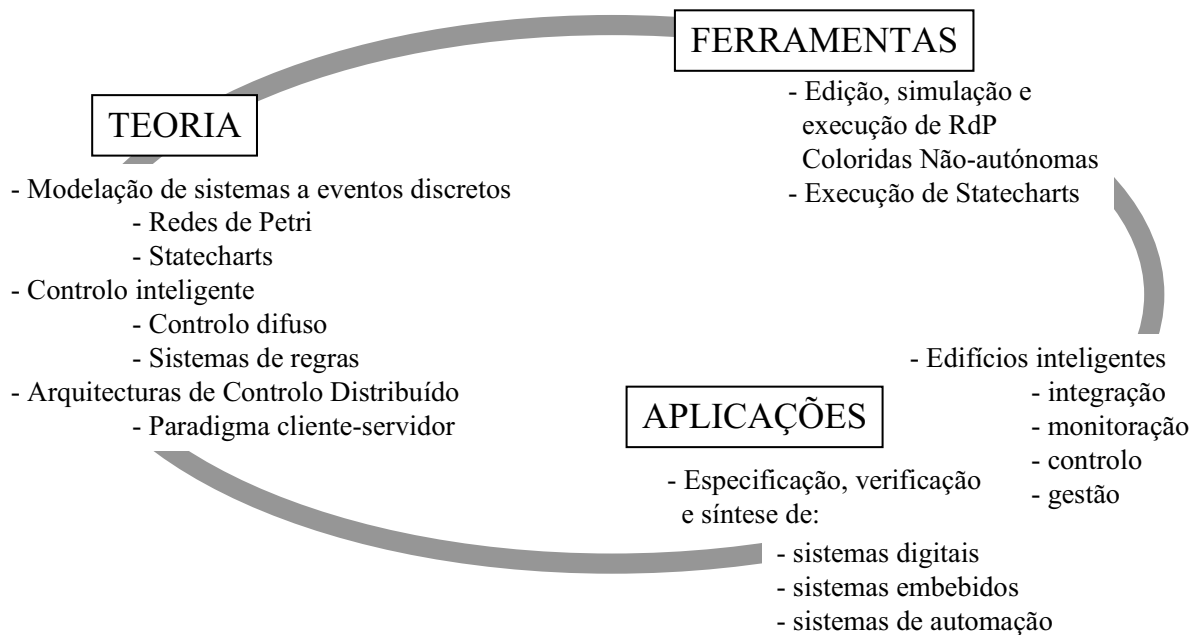


Figura 1-6 - Ciclo de desenvolvimento da metodologia proposta.

Como exemplos significativos de actividades relacionadas com o presente trabalho refiram-se:

- ⇒ ao nível dos conceitos teóricos, identificaram-se duas áreas de referência principais, nomeadamente a dos formalismos gráficos de especificação de sistemas a eventos discretos (com ênfase para as RdP e para os statecharts) e a dos formalismos de especificação de controlo inteligente baseados em paradigmas da Inteligência Artificial e da Inteligência Computacional.
- ⇒ ao nível das aplicações, duas áreas estiveram particularmente presentes: a dos edifícios inteligentes e a de ambientes de apoio ao desenvolvimento de sistemas electrónicos digitais. Enquanto as aplicações a sistemas digitais são comuns na literatura (ver, por exemplo, [Silva, 85] [Fernandes, 94] [Fernandes & Proença, 94] [Fernandes et al., 95] [Adamsky & Monteiro, 95] [Adamsky & Monteiro, 96]), a área dos edifícios inteligentes revelou-se como uma área receptiva a metodologias com capacidade de integrar e suportar os diferentes requisitos e especificidades da panóplia de sistemas presentes ou actuantes nos ambientes prediais. Para além disso, é uma área muito rica do ponto de vista da diversidade dos subsistemas presentes, permitindo um teste constante às capacidades de descrição de sistemas de um formalismo de especificação como as RdP-R e as RdP-RH. Do ponto de vista da especificação do controlo e das estratégias de monitoração e gestão das actividades, torna-se conveniente tentar utilizar, de forma integrada, num elevado número de situações, diferentes formalismos, normas e protocolos. Como exemplos, refiram-se as situações de modelação de controlo de sistemas de climatização (onde a

utilização de controlo difuso se apresenta de elevado interesse), bem como o controlo de múltiplos sistemas a eventos discretos, de que elevadores, controlo de acessos e vigilância são exemplos significativos, para referir só alguns. Todas elas foram aplicações utilizadas para a reflexão sobre os requisitos do modelo de RdP-R e RdP-RH. A utilização de arquitecturas distribuídas complexas recorrendo a redes de comunicação, protocolos e modelos de referência como o de cliente-servidor, tornam a validação do sistema como um todo, numa tarefa ciclópica. As RdP-R e as RdP-RH apresentam-se como formalismos particularmente interessantes para a análise destes tipos de sistemas; contudo, neste particular, não foi possível, por limitações de tempo, proceder à aplicação consistente dos formalismos propostos à modelação dos sistemas distribuídos;

⇒ ao nível das ferramentas, vários foram os protótipos parciais desenvolvidos. Foram desenvolvidos protótipos de ambientes para a edição, análise e simulação da execução de modelos de RdP (preliminares, em relação ao apresentado neste trabalho); face à dimensão dos desenvolvimentos necessários, alguns protótipos preliminares foram utilizados como instrumento de formação em projectos finais de Licenciatura em Engenharia Informática. Complementarmente, para o desenvolvimento de ambientes para a análise de RdP Coloridas e de Statecharts, recorreu-se ao suporte de dois trabalhos realizados no âmbito de teses de Mestrado [Barros, 96] [Soares, 97], e que são dois marcos significativos neste percurso de desenvolvimento de ferramentas.

## **1.5 A estrutura da tese**

A dissertação está organizada em sete capítulos. De uma forma geral, em cada capítulo proceder-se-á à caracterização do estado da arte em relação aos aspectos abordados nesse capítulo. Desta forma, a leitura de cada um dos capítulos pode ser realizada com “alguma” autonomia em relação aos restantes capítulos.

Em seguida, descrevem-se, resumidamente, os conteúdos de cada um dos sete capítulos:

- i) “Sobre os objectivos”, em que se procedeu à identificação dos objectivos do trabalho e dos sistemas a modelar, e onde, de modo claro, se colocou ênfase na atitude de engenharia subjacente ao seu desenvolvimento;
- ii) “As Redes de Petri Reactivas” (RdP-R), onde se procederá à apresentação das características da classe de RdP proposta, partindo das características autónomas das RdP Coloridas, para de forma gradual ir integrando características não-autónomas especialmente adequadas à modelação dos sistemas alvo de projecto. Ao longo do capítulo, sempre que se justifique, identificar-se-ão

modelos comportamentais equivalentes às características propostas, no sentido de deixar claro as técnicas aplicáveis à sua análise;

- iii) “Aspectos de verificação e de implementação”, em que a ênfase se desloca nas questões ligadas à execução e análise das RdP-R. O espaço de estados associado será identificado como única forma geral de análise das RdP-R e será apresentado um algoritmo permitindo a sua construção. Será discutida a utilização de jogadores-de-marcas e do espaço de estados como suporte à execução do modelo, bem como a problemática da resolução automática de conflitos;
- iv) “As Redes de Petri Reactivas e Hierárquicas” (RdP-RH), onde se adicionarão capacidades de decomposição e representação hierárquicas ao modelo não-hierárquico apresentado em capítulo anterior, permitindo a modelação compacta de sistemas de complexidade elevada. Serão propostos três mecanismos básicos de estruturação hierárquica, denominados por decomposição horizontal, vertical e mista, respectivamente. A modelação de statecharts através de RdP Reactivas e Hierárquicas será utilizada como um exemplo das capacidades de implementação de outros formalismos;
- v) “Modelando técnicas de controlo inteligente”, capítulo em que se apresentam as capacidades das RdP-RH de representação de sistemas de regras vários, permitindo a tradução para RdP-RH de vários tipos de formalismos de controlo inteligente, nomeadamente sistemas de regras de produção, sistemas de regras com imprecisão e controladores difusos;
- vi) “Edifícios inteligentes: um conceito potencialmente mobilizador”, onde se procederá à apresentação de propostas para uma abordagem integrada ao conceito de Edifício Inteligente, como um conceito mobilizador; muito embora o ambiente de monitoração e controlo de edifícios a apresentar neste capítulo não tenha ainda recebido o benefício da aplicação directa das RdP-R ou das RdP-RH aos seus sistemas, constituiu-se numa área de aplicação de referência, permitindo testar a aplicabilidade potencial das RdP-R e das RdP-RH, para além de ter sido utilizado como ambiente de validação dos requisitos especificados para os modelos propostos. É um dos ambientes onde se prevê, a curto prazo, a utilização extensiva das RdP-R/RdP-RH a apresentar nesta tese; daí a sua apresentação como um dos seus capítulos;
- vii) “Conclusões”, capítulo dedicado à apresentação das conclusões deste trabalho, bem como à apresentação de algumas linhas potenciais de trabalho futuro.

Da descrição apresentada, identificam-se três partes:

- a primeira, onde se apresentarão as características das classes de RdP a propôr, constituída, grosso modo, pelos primeiros quatro capítulos;

- a segunda onde as características das novas classes serão aplicadas:
  - à tradução de outros formalismos, nomeadamente statecharts e controlo inteligente, apresentado nos capítulos 4 e 5;
  - a uma área de aplicação extremamente diversificada, como a dos edifícios inteligentes, referida no capítulo 6;
- a terceira e última parte, associada ao balanço do trabalho realizado, apresentada no capítulo 7.

## 2. As Redes de Petri Reactivas

### *Sumário do capítulo*

O objectivo do presente capítulo é o de apresentar as características propostas para a classe de Redes de Petri Reactivas, abreviadamente referida por RdP-R. A sua definição decorre dos objectivos enunciados no capítulo anterior, em que os sistemas reactivos de tempo-real foram identificados como os sistemas alvo para modelar e especificar.

Este capítulo é composto pelas seguintes secções:

*2.1 Formalismos para sistemas reactivos de tempo-real*

*2.2 Descrição da classe proposta: as RdP-Reactivas*

*2.3 Análise de um exemplo*

Na primeira secção realiza-se um levantamento dos métodos e ferramentas mais significativos, actualmente disponíveis, aplicáveis à especificação de sistemas reactivos de tempo-real, com ênfase para os que recorrem a uma notação gráfica.

Na segunda secção, as diferentes características das RdP-R são apresentadas. Partindo da classe das Redes de Petri Coloridas, são sucessivamente introduzidas novas capacidades de modelação, permitindo a modelação simultânea dos fluxos de controlo e de dados, de dependências temporais e de sinais de entrada e saída. A arquitectura típica para execução das RdP-R é caracterizada. Os conceitos de ciclo de tratamento, de passos e micropassos de execução (também algumas vezes referidos como de análise) são introduzidos para caracterizar o modo de evolução do modelo.

Finalmente, na terceira secção, recorre-se à modelação simplificada de um subsistema relativo ao protótipo desenvolvido para a monitoração de edifícios, permitindo relacionar as capacidades de modelação de máquinas de estados, statecharts e RdP-R. Evidenciam-se, deste modo, as dependências

já referidas entre conteúdos teóricos (RdP-R, statecharts e arquitecturas cliente-servidor) e de aplicação (edifícios inteligentes).

## 2. As Redes de Petri Reactivas

### 2.1 Formalismos para sistemas reactivos de tempo-real

São inúmeros os métodos disponíveis para a especificação de sistemas, de um modo geral. Enquanto alguns desses métodos dispõem de um forte suporte de ambientes computacionais de desenvolvimento e cobrindo as várias fases do ciclo de vida do sistema, outros são vocacionados para contribuírem predominantemente numa das fases do desenvolvimento, constituindo-se em método de especificação com capacidades de modelar, parcial ou totalmente, os vários aspectos do sistema.

É, desta forma, tarefa complexa e necessariamente incompleta, a de sistematizar a sua enumeração. Na bibliografia podem ser encontradas diferentes taxonomias para a sua classificação; como trabalhos particularmente significativos, refiram-se [Laplante, 93] [Calvez, 93] [Bucci, Campanai & Nesi, 95].

Tomando a taxonomia apresentada em [Bucci, Campanai & Nesi, 95], aplicável aos sistemas de tempo-real de modo geral e dando ênfase à existência de ferramentas computacionais de suporte, identificam-se várias abordagens:

- abordagens operacionais, onde o sistema é descrito através de um modelo executável. O modelo pode ser matematicamente verificado, através de técnicas de análise, e validado através da sua execução (ou simulação);
- abordagens descritivas, baseadas em notações matemáticas, produzindo especificações rigorosas e precisas, permitindo uma caracterização abstracta do espaço de estados associado. O sistema é descrito em termos do *que* deve ser realizado, em vez de *como* deve ser feito (característica das abordagens operacionais);
- abordagens mistas, que tentam beneficiar de características específicas de cada uma das abordagens anteriores.

As abordagens descritivas podem, segundo [Bucci, Campanai & Nesi, 95], ser classificadas em dois grupos, de acordo com a sua característica principal; o primeiro baseado em métodos algébricos (com

recurso ao conceito de Tipos de Dados Abstractos<sup>13</sup> e onde se incluem as linguagens Z, LOTOS e VDM<sup>14</sup>), e o segundo baseado em métodos lógicos (onde se encontram as linguagens RTL<sup>15</sup> e TRIO<sup>16</sup>).

Quer as abordagens descritivas, quer as abordagens mistas não influenciaram as opções tomadas neste trabalho, pelo que não serão analisadas detalhadamente.

As abordagens operacionais podem ser divididas em duas categorias:

- i) na primeira incluem-se os formalismos baseados em modelos orientados por transições, de que as máquinas de estados finitos e as RdP são exemplos; este tipo de modelos é orientado no sentido da descrição do comportamento do sistema, utilizando-se, normalmente, uma representação gráfica;
- ii) na segunda incluem-se os formalismos baseados em notações especialmente vocacionadas para a análise dos sistemas; este tipo de modelos é orientado no sentido da descrição da estrutura ou das funcionalidades do sistema; por essa razão, são, normalmente, encaradas como metodologias de projecto.

No segundo grupo agora referido, encontram-se as metodologias de análise de sistemas comumente utilizadas. Em [Calvez, 93] são divididas em três grupos, de acordo com a fase do ciclo de desenvolvimento a que se encontram associadas:

- predominantemente vocacionadas para a especificação, onde se incluem o SADT - Structured Analysis and Design Technique (Ross), SA - Structured Analysis (De Marco), RTSA - Real-Time Structured Analysis (Ward, Hatley), JSD - Jackson Structured Programming (Jackson), SREM - Software Requirements Engineering Methodology (Alford) e OOA - Object-Oriented Analysis (Coad e Yourdon);
- vocacionadas para o projecto, entendido em sentido lato, onde se incluem o SD - Structured Design (Yourdon e Constantine), JSD - Jackson Structured Programming (Jackson), SYSREM (Alford), DARTS - Design Approach for Real-Time Systems (Gomaa), OOD - Object-Oriented Design (Booch), OMT - Object-Modelling Technique

---

<sup>13</sup> “Tipos de Dados Abstractos”, tradução de “Abstract Data Type”, ADT como normalmente referido.

<sup>14</sup> VDM = Vienna Development Method, produzido nos IBM Laboratories, em Viena, na década de 70.

<sup>15</sup> RTL = Real-Time Logic.

<sup>16</sup> TRIO = Tempo Reale ImplicitO.

(Rumbaugh), HOOD - Hierarchical Object-Oriented Design, SDWMC - System Design With Machine Charts (Buhr).

- vocacionadas predominantemente para a implementação, onde se incluem as metodologias de programação estruturada e orientada por objectos.

Em [Calvez, 93], capítulo 7, pode ser encontrada uma caracterização detalhada destas várias metodologias de análise de sistemas, do ponto de vista dos (aí designados) sistemas embebidos de tempo-real (e neste trabalho classificados como sistemas reactivos de tempo-real).

É, pois, no primeiro grupo de abordagens operacionais referido (modelos orientados por transições), onde a ênfase é colocada em termos da descrição comportamental do sistema, que se encontram as referências significativas, em termos deste trabalho.

A exemplo do que se verifica na generalidade dos formalismos deste grupo, também na classe de RdP a propôr, as RdP-Reactivas, se mantem uma notação predominantemente gráfica, isto é, uma notação gráfica (como comum nas RdP), enriquecida com notações distintivas para as diferentes características do modelo, bem como com anotações textuais com semânticas específicas.

Como tal, as máquinas de estados finitos, os statecharts, a linguagem Esterel e as Redes de Petri serão alvo de uma apresentação individualizada nas subsecções seguintes.

### **2.1.1 Máquinas de estados finitos**

As máquinas de estados finitos (ou autómatos finitos) constituem o formalismo de especificação de sistemas a eventos discretos de utilização mais divulgado. A teoria subjacente remonta à década de 50, com as propostas de Moore e de Mealy. Nas máquinas de Moore, as saídas são associadas aos estados, enquanto que nas máquinas de Mealy, as saídas são associadas às transições (e produzidas como uma função do estado actual e das entradas). A capacidade de descrição dos dois modelos é equivalente, sendo possível a sua tradução mútua.

Três métodos podem ser utilizados para a sua representação [Laplante, 93]: teoria de grafos (recorrendo à teoria de conjuntos), diagramas (modo gráfico) ou através de matrizes. Considerando a legibilidade associada, apenas a representação gráfica se considera de interesse (para efeitos de especificação de sistemas).

De entre as representações gráficas utilizadas comumente, duas são de referência especial: os diagramas de estado e os fluxogramas. Os primeiros são utilizados em imensas áreas de aplicação, enquanto os segundos são vocacionados para a descrição de algoritmos.

Os diagramas de estados são constituídos por círculos representando os estados e arcos representando as transições entre estados. O comportamento do autômato é especificado em termos da dependência das transições entre estados em função de um alfabeto de símbolos de entrada (eventos, do ponto de vista da descrição comportamental do sistema). A Figura 2-1(a) ilustra um exemplo elementar, sendo o estado S1 o estado inicial (indicado, neste caso, por um arco de entrada). Diversas são as propostas existentes para representar máquinas de estado; no exemplo utilizado utiliza-se, implicitamente, o conceito de receptividade a um evento, isto é, nos arcos de saída de um estado só se referem os eventos que provocam transição de estado (como exemplo, no estado S4 apenas existe um arco de saída, modelando uma dependência exclusiva do evento  $r$  para que se obtenha transição desse estado, pelo que os eventos  $a$  e  $b$  não são referidos em arcos de saída do estado S4).

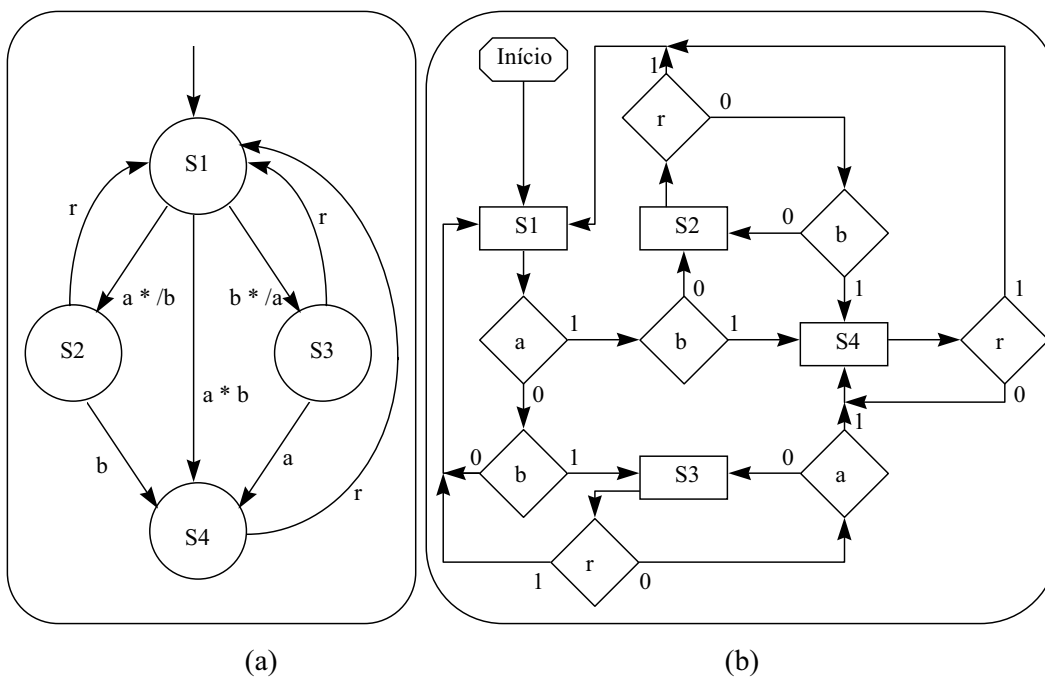


Figura 2-1 - Representação de máquinas de estados finitos:

a) diagrama de estados; b) fluxograma.

No entanto, a especificação apresentada na Figura 2-1(a) não é executável, no sentido em que estão presentes situações de indeterminismo (potenciais conflitos na determinação do estado seguinte). Uma semântica utilizada comumente nos diagramas de estado, permitindo a sua utilização como especificações executáveis, considera a ocorrência mutuamente exclusiva dos eventos especificados; caso isso não seja razoável, torna-se necessário representar explicitamente as dependências existentes (como ilustrado no estado S1 da Figura 2-1(a)), podendo ter como consequência a atribuição de prioridades entre os vários arcos de saída de estado.

Os fluxogramas são, possivelmente, a ferramenta de modelação de *software* mais antiga e podem-se considerar de compreensão intuitiva. Recorrem a representações gráficas em que os estados são

representados por retângulos e as transições entre estados são detalhadas em termos das dependências de eventos de entrada, representadas por losangos, responsáveis pela modelação de testes booleanos. Apresentam o suporte adequado para especificar sistemas a implementar em linguagens como *Assembly* ou *Fortran* (tal como definida décadas atrás), com o inconveniente óbvio de um reduzido suporte às técnicas estruturadas de programação. O estado inicial (e final, quando exista) é explicitamente modelado. A Figura 2-1(b) apresenta um exemplo, correspondente a uma versão executável do diagrama de estados da Figura 2-1(a).

De notar que, enquanto no diagrama de estados da Figura 2-1(a) é possível modelar situações de indeterminismo, no fluxograma da Figura 2-1(b), devido à utilização de testes binários em cascata, garante-se uma especificação determinística. Como exemplo de uma situação de indeterminismo presente na Figura 2-1(a), refira-se a situação de ocorrência simultânea dos eventos  $r$  e  $a$  no estado  $S3$  ou da ocorrência dos eventos  $r$  e  $b$  no estado  $S2$ . No fluxograma associado, a precedência entre os eventos é explicitamente representada, impondo prioridades entre eventos, evitando a situação de indeterminismo.

Como exemplo de uma metodologia particularmente significativa aplicável na área de projecto de sistemas digitais de reduzida complexidade e que recorre à aplicação das técnicas referidas, são de referir os diagramas ASM<sup>17</sup> [Clare, 73]. Em [Treseler, 92] podem encontrar-se técnicas adequadas à implementação de máquinas de estado em circuitos integrados de lógica programável.

Ainda na área de desenvolvimento de sistemas digitais, refiram-se os “*basic schemata*”, desenvolvidos no M.I.T. (ver [Pimenta-Rodrigues & Araújo, 88]) e particularmente vocacionados para a especificação de arquitecturas microcontroladas. Podem ser caracterizados como uma especificação gráfica de algoritmos, não sendo, no entanto, os estados do sistema explicitamente representados.

De um modo geral, as máquinas de estados finitos apresentam limitações sérias à modelação de sistemas complexos, nomeadamente, incapacidade de modelação de actividades concorrentes, bem como de representações hierárquicas tendentes a compactar a representação. Para minimizar essas limitações, várias extensões têm vindo a ser propostas.

Uma extensão comum conduz à utilização de expressões e variáveis auxiliares, embebendo-as na especificação. Deste modo, quer as acções, quer as condições que determinam a transição entre estados, podem ser resultado da avaliação de expressões complexas.

---

<sup>17</sup> Diagramas ASM, tradução de “Algorithm State Machine charts”.

As dificuldades de modelação de processos paralelos e de suportar uma decomposição hierárquica poderão ser consideradas como o calcanhar de Aquiles das máquinas de estados finitos. Várias extensões têm sido propostas para a sua resolução. Os statecharts, a apresentar na subsecção seguinte, constituem uma resposta adequada para a redução destas limitações.

No particular da modelação de concorrência, a máquina de estados equivalente a um conjunto de máquinas de estados com evolução paralela, corresponde a um espaço de estados resultante do produto cartesiano dos espaços de estados de cada uma das máquinas individuais, conduzindo ao fenómeno normalmente referido como de explosão do espaço de estados e a problemas na verificação associada ao modelo resultante. Várias têm sido as propostas apresentadas para a resolução do problema da especificação de máquinas paralelas. Refiram-se, entre outras, as propostas dos CSP (Communicating Sequential Processes) de Hoare e CCS (Calculus of Communicating Systems) de Milner, dos finais da década de 70, bem como as CFSM (Communication Finite State Machine) propostas por Orr e outros na década de 80, executadas através de modelos RdP equivalentes.

### 2.1.2 Statecharts

Os statecharts podem ser caracterizados como uma extensão às máquinas de estados finitos, suportando a sua representação hierárquica, utilizando uma notação gráfica muito intuitiva e permitindo modelar evoluções sequenciais e/ou paralelas, incluindo um mecanismo de comunicação global (“*broadcast*”). Em [Harel, 87] é apresentada a seguinte definição:

statecharts = diagramas de estado + profundidade + ortogonalidade + comunicação global<sup>18</sup>

Cada parcela referida sintetiza uma característica fundamental dos statecharts.

A característica “profundidade” permite implementar uma hierarquia de estados, no sentido em que a cada estado da máquina de estados do nível N, pode estar associada uma máquina de estados no nível N+1; a marcação ou não do estado do nível N permite activar ou desactivar a máquina de estados do nível N+1. O mecanismo de refinamento/abstracção obtido desta forma é caracterizada como sendo de decomposição “ou-exclusivo”, uma vez que se o estado de nível N está activo, apenas um dos estados da máquina de estados associada está activo, e se o estado de nível N não está activo, nenhum dos estados da máquina de estados associada (no nível N+1) o estará. A notação gráfica utiliza um rectângulo arredondado para representar um estado e tira vantagem da utilização de áreas para

---

<sup>18</sup> Tradução de “statecharts = state-diagrams + depth + orthogonality + broadcast-communication”.

representar os estados dependentes dentro da área do estado de nível superior, permitindo uma apreensão intuitiva do comportamento dinâmico especificado.

A característica “ortogonalidade”, também referida por decomposição “e”, suporta a modelação de actividades paralelas.

Por último, a característica “comunicação global” descreve um mecanismo de comunicação instantânea entre as componentes paralelas do modelo considerando o paradigma de tempo de atraso nulo utilizado também pelas linguagens síncronas de especificação de sistemas, nomeadamente pela linguagem Esterel [Berry & Gonthier, 92], a apresentar em subsecção posterior. A comunicação é garantida pela emissão de eventos, associados ao disparo de transições entre estados, que são considerados como eventos de entrada por outras transições.

São, normalmente, consideradas duas semânticas para o atraso nulo na comunicação entre componentes paralelas:

- i) a primeira recorre aos denominados micropassos,
- ii) a segunda recorre ao denominado ponto fixo (ou consistência global, segundo [Beeck, 94]).

No caso de se considerarem micropassos de execução (dentro de um instante ou passo de análise), duas atitudes possíveis são propostas:

- i) na primeira, adoptada pela maioria das variantes, incluindo as propostas de D. Harel, os eventos são considerados unicamente no micropasso seguinte ao da sua activação,
- ii) na segunda, os eventos são considerados persistentes dentro de um passo de execução, isto é, activos para todos os micropassos posteriores ao da sua activação (satisfazendo o princípio da acção-reacção).

Importa referir que, independentemente da semântica considerada, o mecanismo de comunicação instantâneo pode conduzir a contradições, isto é, indeterminismos na execução do modelo [Beeck, 94]. Essa é a situação criada quando se geram ciclos na determinação do estado seguinte, por interacção infinita entre componentes paralelas. A detecção formal dessas situações depende das semânticas utilizadas.

Na Figura 2-2 apresenta-se um exemplo simples mas ilustrando algumas características importantes, extraído de [Harel, 88], em que o statechart da Figura 2-2(a) representa a máquina de estados da Figura 2-2(b) com ganhos claros em termos de legibilidade.

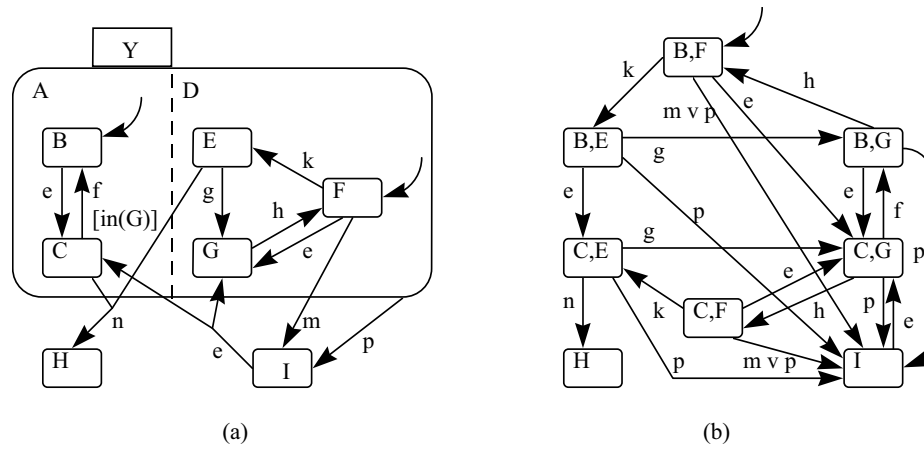


Figura 2-2 - Exemplo de statechart (a) e (b) espaço de estados associado.

Diversas são as semânticas possíveis a considerar para a execução de statecharts. Em [Beeck, 94] é apresentada uma comparação das 20 variantes (consideradas) mais significativas de statecharts; aí se identificam os problemas comumente encontrados nas variantes de statecharts e apresentam-se algumas soluções.

Os statecharts suportam a modelação explícita de interrupção (“*preemption*”) de actividades, nomeadamente a interrupção definitiva (“*abortion*”) e a suspensão. A interrupção é modelada através da utilização de arcos de saída (como os arcos com os eventos  $n$ ,  $m$  e  $p$ , da (a), ligados aos estados  $H$  e  $I$ ). A suspensão é modelada através do conceito de *história*, em que uma capacidade de memória é associada a uma máquina de estados, permitindo retornar ao último estado activo quando de uma nova activação da máquina de estados.

O ambiente de desenvolvimento comercialmente disponível mais divulgado, que recorre a statecharts (não tivesse o “pai” dos statecharts na equipa), é o STATEMATE [STATEMATE, 90], desenvolvido pela iLogix. No STATEMATE consideram-se separadamente os aspectos de modelação estrutural, funcional e comportamental do sistema. Cada um dos aspectos referidos é modelado através de um formalismo específico; os statecharts são, obviamente, utilizados para modelar a componente comportamental.

Como exemplos significativos de trabalhos académicos de implementação de statecharts, refiram-se as Teses de Mestrado [Lucas, 93] e [Soares, 97].

### 2.1.3 Esterel

Esterel é uma linguagem textual que descreve autómatos síncronos que, para além da utilização de instruções elementares como ciclos e testes, dispõe de instruções especiais para impor requisitos temporais [Berry & Cosserat, 85] [Berry & Gonthier, 92].

Em [Chang, Kalavade & Lee, 96] apresenta-se um exemplo elementar, reproduzido na Figura 2-3(a), de um programa com três entradas A, B e R (para reinicialização) e uma saída O, de que a máquina de estados da Figura 2-1(a) é equivalente (excluindo a saída). Na Figura 2-3(b) apresenta-se o statechart equivalente.

O comportamento do programa referido é simples: após o início ou após um sinal de reinicialização, espera até receber os sinais A e B, após o que emite O. Nesse programa podem-se identificar as construções básicas para sequência “;” e para execução paralela “||”. Assim, as instruções “[await A || await B]; emit O;” especificam que o evento O será emitido, apenas depois dos eventos A e B terem ocorrido.

A construção “do ... watching R” permite modelar o mecanismo de interrupção.

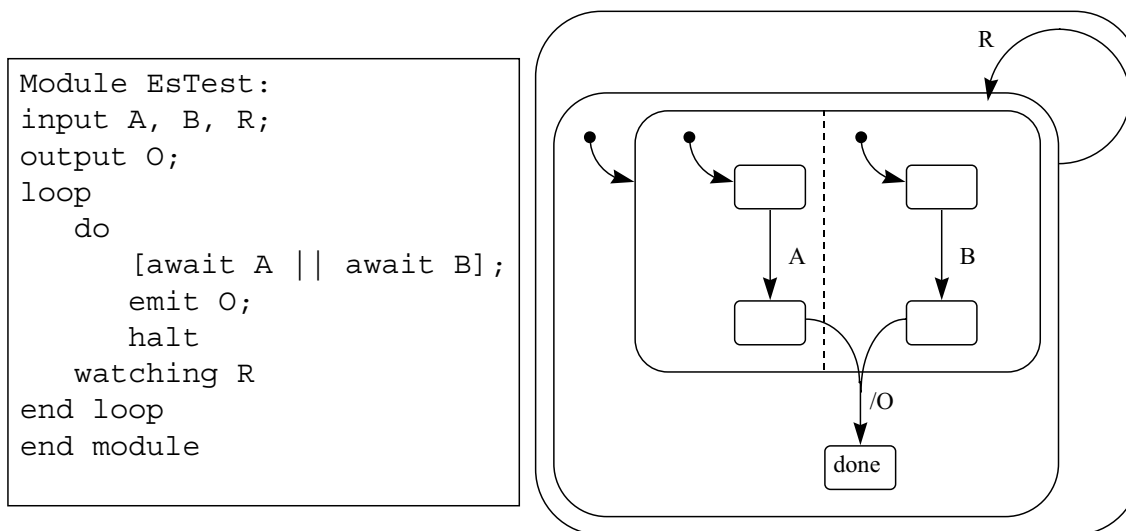


Figura 2-3 - Especificação de um autômato: (a) programa em Esterel; (b) statechart equivalente.

Importa referir que, considerando o exemplo da Figura 2-3, enquanto a dimensão de uma especificação Esterel aumenta linearmente com o número de entradas consideradas (no caso duas), a dimensão da máquina de estados equivalente aumenta exponencialmente, devido ao já referido problema de explosão de estados.

#### 2.1.4 Redes de Petri (RdP)

As RdP foram propostas em 1962 por Carl Adam Petri, matemático alemão, na sua tese de doutoramento. São um instrumento de modelação e análise de sistemas, mais expressivas que as máquinas de estados, que permitem modelar actividades concorrentes e características de sincronização entre processos, exclusão mútua de recursos e memorização. Deste modo, a sua aplicação a numerosas áreas tem vindo a ser realizada. Refiram-se, a título de exemplo, as áreas dos

sistemas distribuídos e as questões de comunicação entre os processos concorrentes, dos sistemas de manufactura e de controlo e gestão de recursos e dos sistemas de automação. Em [Peterson, 77] [Peterson, 81] [Reisig, 82] [Reisig, 92] [Murata, 89] [Silva, 85], entre outros, podem ser encontradas introduções aos fundamentos das RdP, particularmente significativas, bem como introduções às questões da análise das RdP e aplicações a algumas áreas “típicas”. Também em [Tomé, 89], [Gomes, 91a], [Miyagi, 96], [Gomes, 96] e [Cardoso & Valette, 97] se podem encontrar introduções às RdP, em português.

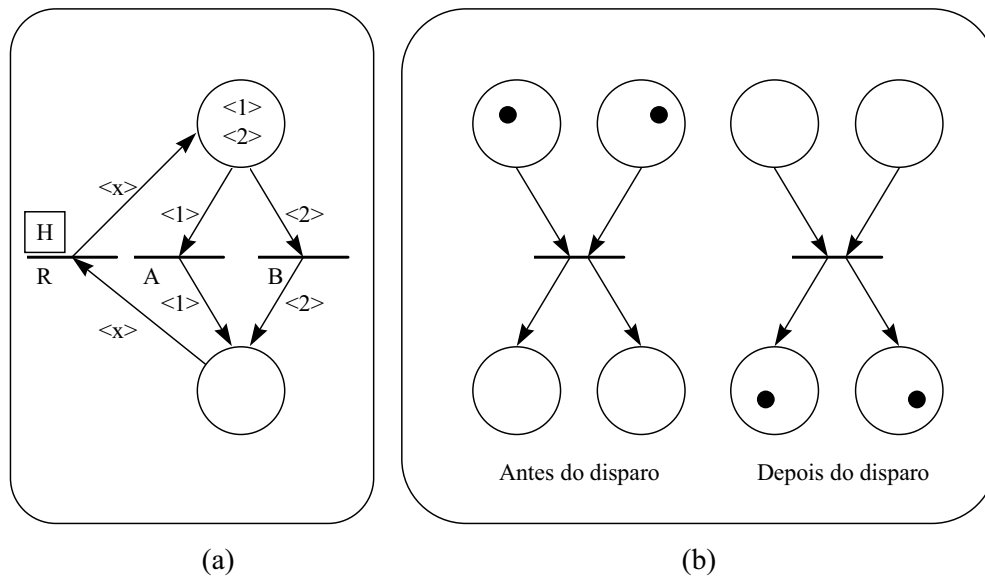


Figura 2-4 - (a) RdP equivalente às especificações da Figura 2-3; (b) Disparo de uma transição.

Embora não se pretenda realizar neste trabalho uma introdução extensa e detalhada às RdP, nos parágrafos seguintes caracterizam-se (muito) brevemente as características mais significativas das RdP em geral, e de algumas classes e extensões com especial interesse para o presente trabalho. Inicia-se a apresentação pelas características autónomas<sup>19</sup> associadas a algumas classes de RdP, seguindo-se algumas extensões que introduzem características não-autónomas nos modelos com RdP. A título introdutório, na Figura 2-4(a) apresenta-se um modelo RdP cujo espaço de estados é semelhante ao das especificações apresentados na Figura 2-3, utilizando Esterel e statecharts.

<sup>19</sup> Por “características autónomas” pretendem-se identificar as características associadas às RdP Autónomas. Por RdP Autónoma entende-se a componente abstracta do modelo RdP do sistema (estrutura do grafo e marcação). Por oposição, os modelos RdP que integram características que o ligam ao “mundo exterior” (por exemplo, sinais de entrada e de saída, temporizações) denominam-se por RdP Não-Autónomas.

### 2.1.4.a RdP Autônomas

Uma RdP pode ser caracterizada por um grafo bipartido, em que os dois conjuntos de nós são denominados por lugares (representados, normalmente, por circunferências ou elipses) e transições (representadas, normalmente, por barras ou retângulos), que são interligados por arcos dirigidos. Um conjunto de marcas pode ser associado a cada um dos lugares. A distribuição das marcas pelos lugares, correspondente ao estado da RdP, denomina-se por marcação da RdP. Estas são as características estáticas da RdP (isto é, a sua estrutura e estado).

A dinâmica associada, isto é, a execução da RdP é realizada através do “disparo” das transições. Uma transição pode disparar (ou ocorrer) quando todos os seus arcos de entrada se interligarem a lugares que se encontrem marcados. A regra de disparo impõe que quando uma transição ocorrer, as marcas presentes nos seus lugares de entrada são consumidas e criadas marcas nos seus lugares de saída. A Figura 2-4(b) ilustra uma situação elementar de disparo de uma transição.

Algumas tentativas de aplicação a novas áreas, tiveram como consequência o aparecimento de novas classes e extensões ao modelo inicialmente proposto. Em [Bernardinello & De Cindio, 92] as várias classes de RdP autônomas são caracterizadas nos seguintes níveis de referência:

- o primeiro nível é caracterizado por lugares marcados por zero ou uma marca sem estrutura associada, isto é, sistemas em que os lugares representam condições;
- o segundo nível é caracterizado por lugares marcados por zero ou várias marcas sem estrutura associada, isto é, sistemas em que os lugares representam contadores;
- finalmente, o terceiro nível é caracterizado por lugares que podem conter marcas com uma estrutura associada; são as normalmente designadas RdP de alto-nível; as classes de RdP associadas aos dois níveis anteriores são, comumente, por oposição, designadas por RdP de baixo-nível.

No primeiro nível incluem-se as RdP Condição-Evento [Reisig, 82], as RdP Elementares [Rozenberg & Thiagarajan, 86] [Thiagarajan, 87], as RdP de Escolha-Livre e as RdP Seguras; alguns autores classificam estas classes como RdP Ordinárias [Silva, 85]. No segundo nível encontram-se as RdP Lugar-Transição; alguns autores referem-se a estas classes como RdP Generalizadas [Murata, 89]. No terceiro nível identificam-se as RdP com Marcas Individuais [Reisig, 92], as RdP Predicado-Transição [Genrich, 86], as RdP Coloridas [Jensen, 86] [Jensen, 90] [Jensen, 92] [Jensen, 95] e as RdP Objecto [Lakos, 95] [Lakos, 96], referindo apenas algumas.

De entre as classes referidas, algumas tentam “isolar” as características elementares “puras” dos vários níveis, como as RdP Elementares e as RdP com Marcas Individuas (referências para os trabalhos mais formais, isto é, para os “matemáticos”), outras tentam enfatizar a sua aplicação a situações de controlo e automação, como as RdP Lugar-Transição, ou integrar as potencialidades da engenharia de software, como as RdP Coloridas e as RdP Objecto.

No sentido de permitir a aplicação de conclusões válidas para uma classe de RdP a outra classe, importa relacionar as várias classes de RdP referidas, permitindo identificar as capacidades de modelação associadas bem como o tipo de propriedades que as diversas técnicas de análise disponíveis permitem determinar. De um modo geral, quanto mais expressiva é a classe de RdP, mais limitadas são as conclusões possíveis pelas técnicas de análise comuns, devido à complexidade inerente.

As classes referidas foram propostas tendo presente esta atitude de possível tradução noutra classe menos expressiva; desta forma, é possível encontrar um modelo RdP Colorida comportamentalmente equivalente a um modelo RdP Objecto, bem como um modelo RdP Lugar-Transição comportamentalmente equivalente a um modelo RdP Colorida, para citar apenas dois exemplos.

#### *2.1.4.b RdP Não-Autónomas*

No entanto, as classes de RdP até agora referidas não são adequadas para modelar alguns aspectos importantes presentes em grande parte de sistemas a eventos discretos, nomeadamente as questões de modelação de tempo e sinais externos. Inúmeros trabalhos têm sido apresentados no sentido de adequar as RdP a situações de aplicação específicas. Alguns desses trabalhos têm conduzido a extensões às RdP, isto é, modelos que, embora mantendo parte da semântica das classes de RdP apresentadas, lhes introduzem características que impedem a sua tradução mútua (não são equivalentes). Será este o caso das RdP-R e RdP-RH a apresentar.

Genericamente, podem ser considerados três tipos de extensões:

- as que conferem uma interpretação específica à rede, permitindo integrar no grafo referências a características do sistema físico (como os sinais de controlo);
- as que conferem a capacidade de testar um estado (qualquer) de marcação;
- as que conferem a capacidade de integrar dependências temporais.

No primeiro grupo encontram-se as RdP Sincronizadas [David, 91] e as RdP Interpretadas [Silva, 85] [David & Alla, 92], em que as necessidades de modelação de controlo determinístico de sistemas

foram o motor de desenvolvimento. Nas RdP Interpretadas definidas em [Silva, 85], as transições dispõem de um evento e de uma condição associados, cuja existência condiciona o seu disparo; sempre que a transição se encontre habilitada (isto é, que todos os seus lugares de entrada possam “fornecer” as marcas de acordo com os arcos a eles ligados), o evento e a condição associados sejam verdadeiros, então a transição dispara obrigatoriamente.

No segundo grupo encontram-se as extensões que permitem ultrapassar a limitação de, numa RdP qualquer, testar o número de marcas presente num determinado lugar. Esta limitação foi caracterizada como a incapacidade de testar o zero. Para o caso particular de um lugar limitado, é possível realizar este teste utilizando o lugar complementar. Por exemplo, se o lugar A for limitado com quatro marcas, o lugar complementar B será inicializado com quatro e o teste "zero marcas em A" poderá ser substituído pelo teste "quatro marcas em B". No entanto, no caso do lugar não ser limitado, o teste de zero continua a não ser possível.

Neste segundo grupo, duas extensões às RdP são de particular interesse: as RdP com arco inibidor e as RdP com prioridades [Hack, 75]. As RdP com prioridades, em que a cada transição se associa uma prioridade relativa, permite, paralelamente, resolver os conflitos eventualmente verificáveis no modelo, suportando, claramente, a execução automática; no caso particular da utilização de circuitos digitais na implementação, o recurso a circuitos decodificadores com prioridade suporta a resolução automática de conflitos.

No terceiro grupo encontram-se extensões que, para além da descrição sobre "o que acontece", introduzem uma dependência temporal. Estas extensões às RdP são particularmente adequadas à análise de desempenho de sistemas, podendo estar na base de simuladores baseados em RdP. Deste modo, surgiram as extensões designadas por RdP temporizadas e com temporizações. De uma forma geral, podem ser consideradas duas atitudes para incluir a dependência temporal:

- ⇒ associando essa dependência aos lugares, obtendo-se RdP temporizadas nos lugares; propõem que uma duração  $d_i$  (variável ou constante, caso particular zero) seja associada a todos os lugares  $p_i$ ; quando uma marca é gerada num lugar, deve aí permanecer durante pelo menos  $d_i$ , ficando, durante esse período não disponível para possíveis disparos de transições;
- ⇒ associando essa dependência às transições, obtendo-se RdP temporizadas nas transições; propõem que uma duração  $d_i$  (variável ou constante, caso particular zero) seja associada a todas as transições  $t_i$ ; neste caso, uma marca deverá ser reservada durante um intervalo  $d_i$  para permitir o disparo de uma transição; apenas as marcas não reservadas poderão ser consideradas para habilitar uma transição.

Os dois tipos apresentados poderão ser convertidos reciprocamente, pelo que a selecção da abordagem mais correcta depende da aplicação.

Como caso particular, de elevado interesse para a avaliação de desempenho de sistemas, refiram-se as RdP Estocásticas, propostas em [Natkin, 80] na sua dissertação de doutoramento, e as RdP Estocásticas Generalizadas [Marsan et al., 84], em que se consideram durações aleatórias com distribuição exponencial associadas às transições.

Considerando as propostas de classes de alto-nível, novas formas de modelar o tempo foram propostas, nomeadamente aquelas que associam temporizações aos arcos e às marcas. Do ponto de vista formal, correspondem a trazer para essas classes, as dependências temporais propostas anteriormente associadas a transições e a lugares.

De particular interesse, refiram-se as RdP Coloridas Temporizadas [Jensen, 95], em que se utiliza um atributo de cor especial que, associado a cada marca e dependendo de um relógio global sempre crescente, determinará a disponibilidade da marca para ser envolvida no disparo de uma transição. As temporizações são determinadas por expressões de arco, de forma semelhante a qualquer outro atributo de cor.

#### *2.1.4.c Representação de arcos*

A título de considerações finais nesta subsecção, aborda-se uma questão que se considera de primordial importância para a utilização generalizada das RdP. É a questão da notação gráfica utilizada e da capacidade de representar de forma distintiva diferentes semânticas associadas aos arcos das RdP. Tomando as RdP Coloridas como referência (embora as conclusões possam ser aplicáveis, obviamente, a RdP de baixo-nível), em [Lakos & Christensen, 94] (parte final) são apresentados os trabalhos relevantes propondo extensões a arcos. Nesse trabalho são propostos vários tipos de arcos, simples e compostos, bem como a semântica associada à sua utilização. Na Tabela 2-1 são apresentadas as propostas de [Lakos & Christensen, 94] relativas a arcos simples (arcos de entrada e saída, de teste e inibidor) e alguns arcos compostos (reserva, igualdade, remove e inicializa). As propostas relativas aos arcos simples foram inicialmente apresentadas em [Christensen & Hansen, 93].

| Tipo     | Grafismo | Nome       | Condição de habilitação                 | Efeito da ocorrência                         |
|----------|----------|------------|---|--|
| Simples  |          | Entrada    | $m(p) \geq \text{exp}\langle b \rangle$ | $m'(p) = m(p) - \text{exp}\langle b \rangle$ |
|          |          | Saída      | -                                       | $m'(p) = m(p) + \text{exp}\langle b \rangle$ |
|          |          | Teste      | $m(p) \geq \text{exp}\langle b \rangle$ | $m'(p) = m(p)$                               |
|          |          | Inibidor   | $m(p) \leq \text{exp}\langle b \rangle$ | $m'(p) = m(p)$                               |
| Composto |          | Reserva    | $m(p) \geq \text{exp}\langle b \rangle$ | $m'(p) = m(p)$                               |
|          |          | Igualdade  | $m(p) = \text{exp}\langle b \rangle$    | $m'(p) = m(p)$                               |
|          |          | Remove     | $m(p) = \text{exp}\langle b \rangle$    | $m(p) = \emptyset$                           |
|          |          | Inicializa | $m(p) = \emptyset$                      | $m(p) = \text{exp}\langle b \rangle$         |

Tabela 2-1 - Tipos de arcos e semântica associada.

A integração progressiva, a partir de modelos simples (iniciando-se nas RdP Coloridas), dos vários tipos de arcos é realizada sistematicamente, provando a sua possível tradução em termos de RdP Coloridas comportamentalmente equivalentes (ver [Christensen & Hansen, 93] e [Lakos & Christensen, 94]). Por exemplo, o arco de teste é comportamentalmente semelhante ao arco de reserva (“vulgar” ciclo) em termos das suas condições de habilitação e de disparo; no entanto, o primeiro não cativa as marcas envolvidas, podendo uma mesma marca ser envolvida em vários arcos de teste e um arco com reserva simultaneamente.

A exemplo do que ocorre nas RdP de baixo-nível limitadas, o arco inibidor pode ser traduzido por um arco de teste em relação ao lugar complementar do lugar onde está ligado (através da transformação de um lugar  $p$ , da marcação inicial e da capacidade que lhe estão associadas para um par lugar-lugar complementar,  $p$ - $p'$ , em que o lugar complementar  $p'$  será inicialmente marcado com o multiconjunto complementar da marcação inicial de  $p$  em relação à capacidade do lugar  $p$  [Christensen & Hansen, 93]).

## **2.2 Descrição da classe proposta: as RdP-Reactivas**

O modelo proposto neste trabalho como classe de referência, denominada por Redes de Petri Reactivas e referidas abreviadamente por RdP-R, tenta integrar algumas das características identificadas nalguns dos formalismos apresentados, particularmente adequadas quando consideradas as necessidades específicas de modelação dos sistemas e objectivos propostos. Um número significativo dessas características foram já apresentadas em [Gomes & Steiger-Garção, 92], [Gomes & Steiger-Garção, 95d], [Gomes & Steiger-Garção, 96b] e [Gomes & Steiger-Garção, 96c].

A apresentação das diversas características das RdP-R será realizada de modo incremental, partindo da identificação das RdP Coloridas como classe base de referência e através da adição sucessiva de novas características, quer em termos da modelação da parte não-autónoma do modelo e da identificação de semânticas específicas utilizadas, quer em termos dos condicionantes que a arquitectura de execução deverá satisfazer. Em capítulo posterior, estas características serão complementadas com capacidades de representação hierárquica, dando origem às RdP Reactivas e Hierárquicas, referidas abreviadamente por RdP-RH.

A proposta destas novas classes de RdP, enquadrada pelos objectivos expostos em capítulo anterior, permitirá encontrar uma ferramenta de modelação suficientemente flexível (no que respeita ao seu modo de utilização pelo modelador de sistemas) e poderosa, particularmente adequada para a modelação de sistemas complexos (incluindo as partes de controlo e de processamento de dados) em que se pretendem incluir dependências temporais.

As características “fortes” são herdadas da integração das características das RdP Coloridas [Jensen, 92], das RdP Interpretadas [Silva, 85] e dos statecharts [Harel, 87], permitindo obter modelos compactos (no sentido utilizado nas RdP Coloridas), organizados hierarquicamente (no sentido utilizado nas RdP Coloridas e nos statecharts) e com capacidades de modelação de dependências de sinais de entrada e saída e de temporizações (no sentido utilizado nas RdP Interpretadas e nos statecharts).

As capacidades de modelação oferecidas pelos três formalismos referidos (RdP Coloridas, RdP Interpretadas e statecharts) podem ser consideradas como casos particulares (subconjuntos) das capacidades disponibilizadas pelas classes a propôr. Transfere-se para o projectista a utilização ponderada dessas várias características, em face da aplicação pretendida.

A intenção de utilizar o (mesmo) modelo nas tarefas de especificação, validação, verificação e implementação impõe que o modelo seja executável; na presente secção serão definidas as semânticas das características propostas necessárias para atingir esse objectivo.

Importa referir que para a prossecução dos objectivos enunciados em capítulo anterior, considera-se estritamente necessário dispôr, de modo integrado, das capacidades de modelação disponibilizadas pelos três formalismos atrás referidos. Não se conhecendo nenhum formalismo com essas características, propõem-se as RdP-R e RdP-RH, proposta que neste trabalho se concretiza e explora.

Como comentário final antes da apresentação individualizada das várias características das RdP-R, refira-se que os resultados conhecidos associados aos três formalismos referidos podem ser integrados directamente na análise de modelos RdP-R e RdP-RH. No capítulo seguinte este tema será retomado.

### 2.2.1 Parte autónoma do modelo

A parte autónoma das RdP-R tem como referência as RdP Coloridas [Jensen, 92]. Do ponto de vista da definição formal destas características não se verificam diferenças em relação às características semelhantes apresentadas na bibliografia específica das RdP Coloridas. Esta escolha é motivada pelas necessidades de modelação identificadas para os sistemas reactivos de tempo-real, em que são necessárias capacidades de compactação do modelo para gerir convenientemente a sua potencial complexidade.

São as seguintes as características estáticas das RdP-R, herdadas das RdP Coloridas (ver [Jensen, 92] pp. 69 a 73):

- é um grafo bipartido, contendo arcos dirigidos e cujos nós podem ser lugares e transições;
- define-se um conjunto de cores, como sendo tipos de dados;
- a cada lugar está associada uma cor, representando o tipo de marcas que o lugar pode conter, isto é, as marcas devem satisfazer as cores definidas;
- a marcação de cada lugar é um multiconjunto de marcas (admitindo-se, portanto, repetições);
- cada arco do grafo pode dispôr de uma inscrição associada, representando a marca transportada e cujo tipo é igual à cor do lugar a que o arco está interligado;
- cada transição pode dispôr de uma guarda, impondo restrições às características das marcas transportadas pelos arcos de entrada a ela ligados;
- está definida uma marcação inicial da rede.

A notação gráfica utilizada neste trabalho para as RdP-R representa as transições por barras e os lugares por elipses (ou circunferências). Também a indicação da marcação inicial pode ser alvo de uma notação distintiva, caso a complexidade do modelo o justifique e no sentido de evitar confusão com outras características textuais presentes (nomeadamente entradas e saídas, a referir posteriormente). Propõe-se a utilização de um par de linhas contendo a indicação das características da marcação inicial ligado ao lugar respectivo através de um arco. A notação gráfica alternativa, também possível e comumente utilizada, recorre à indicação da marcação inicial dentro ou próximo do lugar respectivo.

Do ponto de vista das características dinâmicas, isto é, da execução da RdP-R, considera-se que uma transição está habilitada a disparar em relação a um determinado vínculo<sup>20</sup> quando existe um conjunto de marcas presentes nos lugares de entrada da transição em número e tipo coerentes com as inscrições dos arcos de entrada e da guarda da transição em causa (de acordo com [Jensen, 92] pp. 73 a 78); nessa situação, as marcas dos lugares de entrada envolvidas são destruídas e, simultaneamente, criadas marcas nos lugares de saída de acordo com as inscrições dos arcos de saída.

Como demonstrado em [Jensen, 92] pp. 78 a 85, é possível obter um modelo RdP Lugar-Transição comportamentalmente equivalente a um modelo RdP Colorida. Para garantir o regresso às raízes, também as RdP Lugar-Transição podem ser traduzidas em RdP Condição-Evento (propostas, inicialmente, pelo Professor Petri) comportamentalmente equivalentes [Desel & Merceron, 89].

## 2.2.2 Caracterização da arquitectura de execução

Considerando que o motor do presente trabalho endereça a especificação de controlo para sistemas reactivos de tempo-real, importa definir uma arquitectura de referência para o controlador a utilizar, ou seja, para a plataforma a utilizar na execução do modelo a especificar.

Como referido, a caracterização do sistema é realizada em termos de um modelo comportamental recorrendo às RdP-R (posteriormente complementado com as RdR Reactivas e Hierárquicas).

A estrutura de controlador proposta é apresentada na Figura 2-5. Esta estrutura é uma “versão integrada” resultado das contribuições dos tradicionais controladores digitais, controladores lógicos programáveis e dos controladores difusos, adaptada de [Gomes & Steiger-Garção, 92] [Gomes & Steiger-Garção, 95a] [Gomes & Steiger-Garção, 95d].

---

<sup>20</sup> “Vínculo”: utilizado como tradução de “binding”.

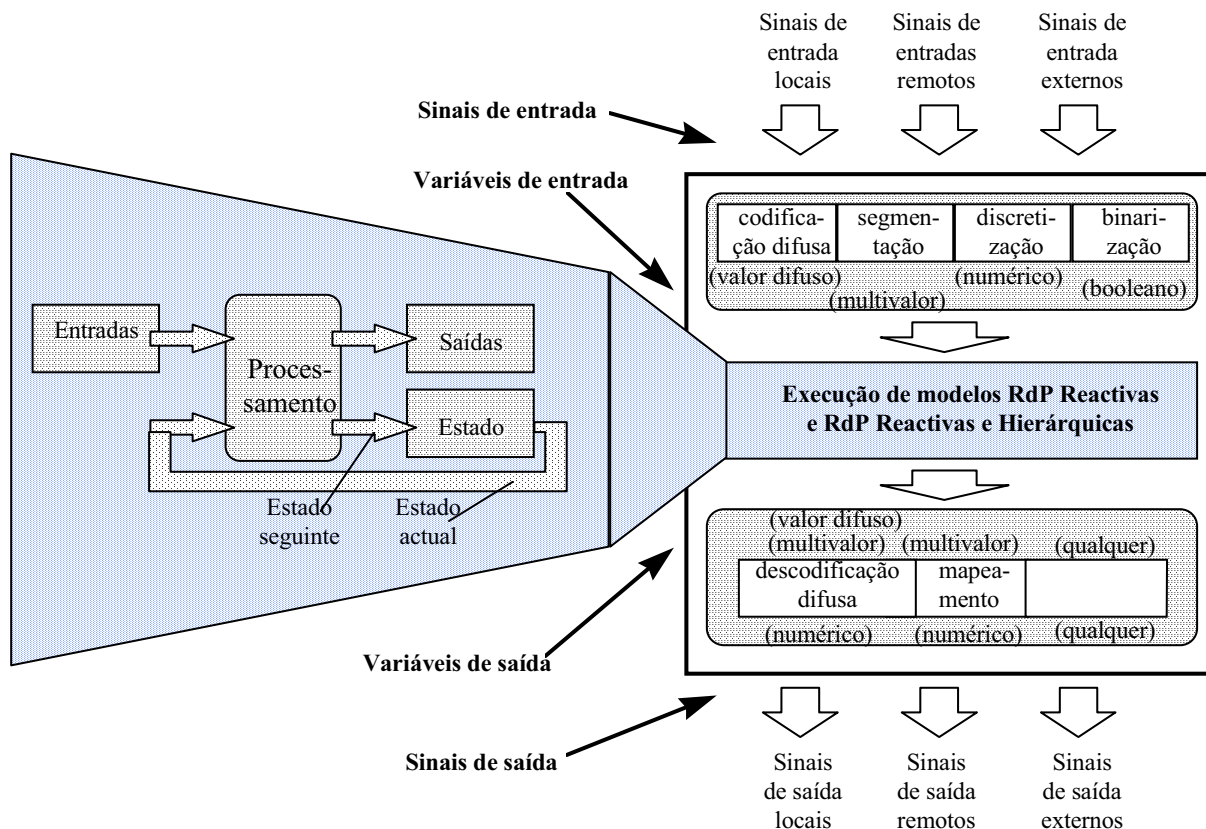


Figura 2-5 - Estrutura de referência do controlador.

Consideram-se diferentes tipos de sinais de entrada e saída, divididos em três grupos, de acordo com a sua localização física e tipo de interligação ao controlador, nomeadamente:

- sinais locais, sob controlo directo do controlador;
- sinais remotos, controláveis através de subsistemas interligados ao controlador;
- sinais externos, controláveis através de outros controladores, acessíveis através de redes de comunicação.

A distinção entre entradas e saídas físicas e a sua representação computacional é realizada denominando os primeiros por sinais e a representação simbólica associada, por variáveis. A distinção é realmente significativa apenas para os tipos de sinais não elementares (difuso e multivalor, a apresentar posteriormente). Para os tipos de sinais elementares as duas classificações são utilizadas indistintamente.

Do ponto de vista do modelo RdP-R, os vários tipos de sinais são referidos de forma indistinta. Os procedimentos específicos para lhes ter acesso são dependentes das plataformas de suporte à sua execução. Como exemplos, utilizações típicas dos sinais externos incluem o suporte a aplicações

distribuídas e comunicações com sistemas SCADA<sup>21</sup> e interfaces de utilizador; por seu lado, os sinais locais e remotos estarão, normalmente, ligados directamente ao processo a ser controlado.

O coração da estrutura proposta é responsável pela execução do modelo RdP-R, suportando a integração de formalismos distintos de especificação de controlo, como referido em capítulo anterior.

De um modo geral, as variáveis utilizadas por um controlador podem ser classificadas nos seguintes tipos [Albertos, 92] [Gomes & Steiger-Garção, 95a]:

- binárias (ou booleanas), podendo assumir um de dois valores;
- multivalor, em que o domínio de variação possível é segmentado em várias zonas, recebendo cada zona um identificador único;
- estocástico, quando a caracterização do sinal de entrada é uma variável aleatória estocástica;
- difuso, quando o sinal é caracterizado por um número de conjuntos difusos, identificados por valores linguísticos;
- discretas, quando a caracterização da amplitude do sinal é realizada por uma sequência de valores;
- contínuas, para os sinais que dependem do tempo (de modo contínuo).

Os últimos dois tipos enfatizam as dependências temporais dos sinais de modo tal que ficam fora das caracterizações de interesse para sistemas a eventos discretos.

Importa referir que as necessidades actuais de controlo de processos, ao colocarem novos desafios às técnicas de controlo, aliadas às capacidades de processamento suportadas pelas actuais tecnologias, tendem a considerar como sinais de interesse, para além dos já identificados, outros de natureza mais complexa.

Como exemplo significativo de trabalhos em curso neste enquadramento, refira-se o projecto ESPRIT 21017 INNOVA - “High Performance Telecontrol Station with Image and Sound Capabilities” (no qual o autor do presente trabalho participa e é responsável pela coordenação de dois dos onze *workpackages*), cujo objectivo é o de especificar e desenvolver uma estação de telecontrolo e telemedida para sistemas de tempo-real que, para além de considerar sinais convencionais, como os

---

<sup>21</sup> “SCADA = Supervisory, Control and Data Acquisition Systems”.

sinais binários e analógicos, se processam sinais de som e de imagem (não do ponto de vista de uma estação multimédia, mas do ponto de vista de controlo industrial). No presente trabalho não são considerados estes tipos de sinais. Constitui, no entanto, uma área de trabalho de elevado interesse, que se tentará desenvolver no futuro próximo, para a qual não se conhecem formalismos de especificação e verificação completamente satisfatórios e em relação à qual a aplicação das RdP-R se apresenta com elevado potencial de adequação.

Neste trabalho são, pois, considerados os quatro primeiros tipos atrás referidos, nomeadamente binários, multivalor, estocásticos e difusos.

O primeiro tipo está associado aos sinais lógicos binários, os mais comuns, normalmente utilizados em sistemas digitais e de automação. Serão referidos na sua representação booleana através de “0” e “1”, podendo ser utilizada a representação simbólica de “verdade-falso” ou “ligado-desligado”. A sua representação interna utiliza um bit.

Os restantes três tipos (multivalor, estocástico e difuso) estão associados a sinais analógicos ou representados através de um código binário; necessitarão de ser processados por conversores analógico-digitais.

A aquisição de variáveis multivalor pode ser realizada através de processos software ou hardware. Uma variável de tipo multivalor corresponde a um tipo enumerado; a sua representação interna será realizada através de uma palavra com  $n$  bits, dependendo da sua cardinalidade e do código utilizado. No processo de aquisição, tabelas de índice ou circuitos implementados com comparadores podem ser utilizados, enquanto que no processo de activação se podem utilizar conversores digital-analógicos responsáveis pela produção de determinados níveis em função do código utilizado, isto é, do mapeamento de códigos em valores analógicos.

Um sinal difuso, associado com uma variável difusa ou linguística, possui associado um número (normalmente pequeno) de valores linguísticos que são interpretados como identificadores de conjuntos difusos, aos quais se associam funções de pertença difusa. A sua representação interna é composta por um conjunto de  $m$  tuplos, cada um dos quais associado com um valor linguístico. Cada tuplo é composto por uma palavra com  $n$  bits, representando o identificador, e por uma informação numérica codificando o valor de pertença. A aquisição de uma variável linguística pode ser realizada através de um processo hardware ou software; a caracterização e a implementação pode ser realizada tendo em conta um método específico de codificação difusa (também referido nalguma literatura por

conversão linguístico-numérica, mas normalmente designado por “fuzzificação<sup>22</sup>”), ver [Lee, 90], por exemplo. No processo de obtenção de um sinal a partir da variável linguística associada, diversos métodos de “defuzzificação<sup>23</sup>” ou de descodificação difusa podem ser utilizados [Lee, 90]; entre os exemplos mais comuns incluem-se os critérios de máximo, de média de máximos ou de centro de massas.

A aquisição e caracterização de sinais afectados por ruído aleatório pode ser realizada tomando uma de três atitudes:

- utilizar o seu valor numérico instantâneo;
- utilizar uma caracterização de variável multivalor, através da partição booleana do domínio do sinal; a caracterização probabilística do sinal de entrada permite obter a segmentação do espaço de entrada; alguns trabalhos na área de aprendizagem automática permitem obter os métodos adequados para essa caracterização [Moura-Pires, 93];
- utilizar uma caracterização de variável difusa, através da partição difusa do domínio do sinal; diferentes métodos podem ser utilizados para proceder à codificação difusa [Lee, 90] baseados em informação probabilística; as funções de pertença podem ser determinadas através de vários métodos, nomeadamente:
  - semelhantes às funções de densidade de probabilidade;
  - definidas por um triângulo isósceles, em que o vértice coincide com o ponto de média da informação recolhida e a base é igual a duas vezes o desvio padrão da distribuição;
  - considerando as propostas de Dubois e Prade, em que a função de pertença é inferida a partir de um histograma obtido pela transformação da medida de probabilidade numa medida de possibilidade, utilizando o conceito de grau de necessidade.

---

<sup>22</sup> Resultado da adaptação directa do termo “fuzzification”. Lamentavelmente, expressões alternativas, que permitiriam uma tradução mais tranquila, não se têm imposto. Como exemplo, refira-se “fuzzy encoder”, proposto por W. Pedrycz, permitindo a tradução para “codificador ou conversor difuso”.

<sup>23</sup> Neste caso utilizou-se “descodificação difusa” como tradução para “defuzzification”, comumente realizada através de “defuzzificação” ou de “conversão numérico-linguística”, como também utilizado.

A representação interna associada aos valores numéricos referidos depende do tipo escolhido para a representação, sendo possíveis os tipos elementares normalmente utilizados para representar informação, nomeadamente inteiros e reais.

A Figura 2-6 apresenta a relação entre os quatro tipos de sinais e suas representações internas, em termos das técnicas de codificação aplicáveis. Para simplificação, na Figura 2-6, uma representação interna de tipo numérico pode corresponder a qualquer dos tipos de sinais apresentados.

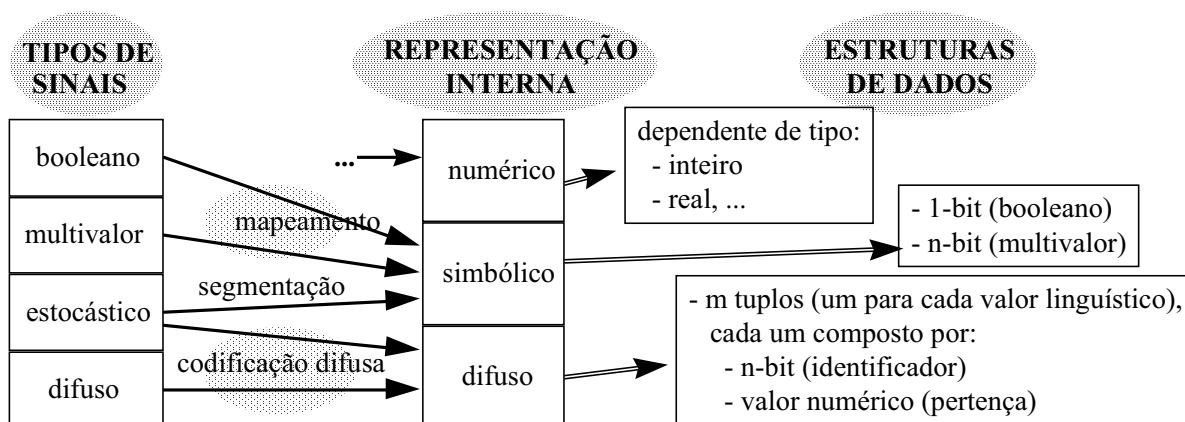


Figura 2-6 - Tipos de sinais, sua representação interna e estruturas de dados associadas.

Considerando o tipo de sistemas que se pretendem modelar, é possível utilizar vectores de sinais (uni- ou multidimensionais) associados a um mesmo identificador, de modo a melhorar quer a legibilidade, quer o nível de compactação do modelo.

### 2.2.3 Modelação de entradas

Os sinais de entrada são modelados através de condições deles dependentes associadas às transições. Do ponto de vista da caracterização formal, cada transição tem associado um evento (como proposto em [David, 91] para as RdP Sincronizadas e em [Silva, 85] para as RdP Interpretadas) ou condição lógica envolvendo um ou vários eventos. Como caso particular, refira-se a possibilidade da condição permitir a avaliação da negação de um evento de entrada.

Do ponto de vista da notação gráfica utilizada, recorre-se à utilização de uma seta (só contorno) “a entrar” num dos extremos da barra representativa da transição, à qual se associa a dependência nos eventos externos (ver, por exemplo, Figura 2-7(a)). Alternativamente, caso a complexidade do modelo o justifique, utiliza-se um hexágono contendo a expressão e ligado à transição através de um arco terminando numa seta (só contorno).

As regras de disparo de transição atrás apresentadas devem ser alteradas no sentido de integrar a dependência dos eventos agora referida. Deste modo, o disparo de uma transição ocorre sempre que:

- esteja habilitada do ponto de vista das características autónomas do modelo, isto é, exista um vínculo possível no estado de marcação;
- esteja pronta do ponto de vista das características não-autónomas, isto é, a condição associada dependente de eventos seja avaliada como verdadeira.

Um aspecto complementar refere-se à legibilidade associada à representação dos eventos de entrada, isto é, sobre quanto a notação gráfica utilizada é adequada para a representação compacta das dependências nos eventos de entrada. Quando se pretende modelar uma situação de conflito estrutural (em termos da componente autónoma do modelo), situação em que é a avaliação de condições em termos de eventos de entrada que determina a evolução efectiva, torna-se de elevado interesse a disponibilidade de uma representação compactada para modelar essas dependências. É uma situação comum quando se consideram áreas de aplicação como as endereçadas pelo presente trabalho.

No sentido de aumentar a legibilidade e a capacidade expressiva das RdP-R, propõe-se a utilização de uma representação compactada para modelar os diferentes fluxos possíveis de evolução dependentes da indicação de eventos de entrada.

Propõem-se duas construções específicas:

- a primeira permite modelar a avaliação como falsa da condição associada a uma transição; do ponto de vista algorítmico, permite modelar estruturas de controlo do tipo IF-THEN-ELSE; a notação gráfica proposta utiliza uma barra adicional, como ilustrado na Figura 2-7(b), em que a avaliação do evento  $a$  como verdadeiro provoca a transição do lugar S3 para S5, enquanto que se o evento  $a$  for avaliado como falso, a transição permitida é do lugar S3 para S4;
- a segunda permite modelar a avaliação multi-valor de uma expressão; do ponto de vista algorítmico, permite modelar estruturas de controlo do tipo CASE; a notação gráfica proposta utiliza um arco de circunferência com arcos de saída associados às várias condições que se pretendem avaliar, consideradas mutuamente exclusivas; a Figura 2-7(c) ilustra uma situação de modelação de três fluxos possíveis, associados à avaliação das expressões 'a=1', 'a=2' e 'a=3, respectivamente.

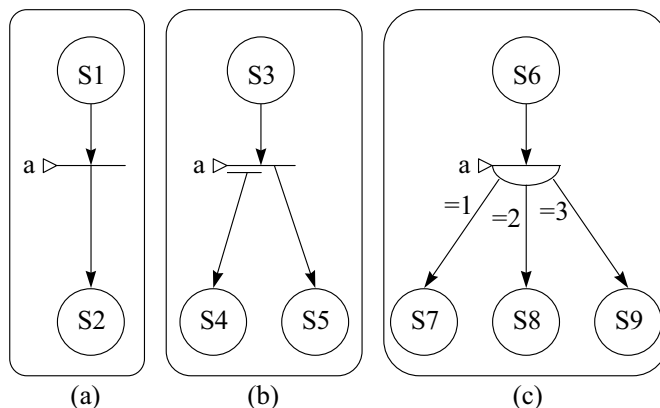


Figura 2-7 - Compactação da representação do fluxo de controlo.

A limitação imposta à ocorrência mutuamente exclusiva dos eventos associados às múltiplas evoluções especificadas (ver Figura 2-7(c)) garantindo o determinismo na execução do modelo, pode ser retirada permitindo a modelação de situações de não-determinismo desde que se considere a utilização de mecanismos de resolução de conflitos, como os apresentados em capítulo posterior deste trabalho.

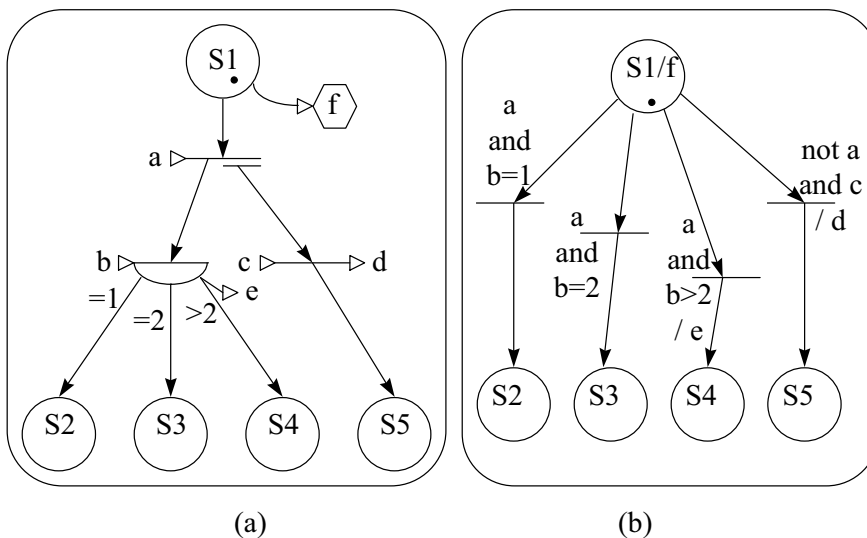


Figura 2-8 - Representação compacta e modelo RdP equivalente.

Estas construções podem ser utilizadas em cascata, como normalmente utilizado em formalismos gráficos de representações de fluxo de controlo, como é o caso dos fluxogramas. De notar que, esta forma de representação continua a garantir a característica de grafo bipartido das RdP, dado que a especificação é “pré-processada” no sentido de obter a RdP “comum” subjacente. A Figura 2-8 ilustra a aplicação da notação a uma situação básica, apresentando-se a forma compacta (com legibilidade acrescida) e o modelo RdP equivalente.

Realça-se que, para manter a coerência do processo de tradução, a indicação das expressões de arco, associadas às características coloridas do modelo, devem ser realizadas unicamente nos arcos directamente ligados a lugares. No tocante às guardas a utilizar na RdP equivalente, são resultantes da conjunção das guardas utilizadas no percurso equivalente na representação compacta.

#### **2.2.4 Sincronismo em relação a eventos externos**

No sentido de garantir um comportamento determinístico como resultado da execução de uma especificação, importa sacrificar o não-determinismo, intrínseco às RdP, no disparo das transições habilitadas. Adopta-se a proposta das RdP Sincronizadas [David, 91] e das RdP Interpretadas [Silva, 85], em que o disparo da transição se verifica sempre que a transição se encontre habilitada (do ponto de vista das características autónomas) e pronta (do ponto de vista das características não-autónomas). É, de resto, a atitude normalmente encontrada nas situações de aplicação à modelação de controlo de sistemas, nomeadamente sistemas de automação e sistemas digitais.

A adopção desta disciplina de disparo imediato das transições habilitadas e prontas, coloca, com ênfase especial para os objectivos perseguidos, as questões associadas à resolução dos conflitos entre disparos de várias transições habilitadas por uma mesma marca. Com efeito, sendo a execução autónoma do modelo um dos objectivos em vista, a resolução automática de conflitos reveste-se de uma importância significativa. Em capítulo posterior retomar-se-á o tema.

#### **2.2.5 Utilização de prioridades**

No sentido de suportar a resolução automática de conflitos, as transições podem dispôr de um atributo de prioridade. Esta característica também será nuclear para suportar as técnicas de estruturação hierárquica utilizadas nas RdP-RH. Ambos os temas serão abordados em capítulos posteriores.

Como frequentemente identificado na literatura, a utilização de prioridades associadas às transições permite capacidades de modelação equivalentes às associadas a máquinas de Turing [Peterson, 77].

Como contra-partida ao acréscimo das capacidades de modelação, refira-se a redução no número de propriedades do modelo caracterizáveis através dos métodos de análise comuns.

Não é, no entanto, esta uma limitação significativa, dado que paralelamente se pretende associar características não-autónomas às RdP-R, que se traduzem numa limitação semelhante à aplicabilidade de algumas técnicas de análise, nomeadamente as técnicas baseadas na determinação e análise de invariantes.

A consequência, em termos dos métodos de análise utilizáveis, pode resumir-se a uma dependência absoluta nas técnicas de análise do espaço de estados associado. Será, portanto, a análise do espaço de estados que permitirá determinar todas as características de um determinado modelo, nomeadamente vivacidade e recursos necessários (por exemplo, capacidades de lugares).

O recurso à utilização de prioridades será frequente ao longo deste trabalho, sendo uma característica importante para a resolução automática de conflitos e para algumas técnicas a propôr para a estruturação hierárquica do modelo, permitindo definir as Redes de Petri Reactivas e Hierárquicas, a apresentar em capítulo posterior e abreviadamente representadas por RdP-RH.

Admitindo que dois níveis de prioridade permitem modelar a maior parte das situações de interesse, é, normalmente, suficiente associar um atributo que pode tomar o valor “alta” ou “baixa”, representado abreviadamente por “H” ou “L”. Em situações específicas, cada um dos valores pode ser subdividido, utilizando um inteiro cujo valor reforça a característica. Isto é, uma transição com o atributo “alta3” (“baixa3”) será mais (menos) prioritária do que outra com o atributo “alta2” (“baixa2”, respectivamente). Do ponto de vista da notação gráfica utilizada, as prioridades são representadas dentro de um rectângulo junto à transição.

### **2.2.6 Evolução do estado do modelo**

Considera-se que a evolução do estado das RdP-R é unicamente possível em instantes especiais de análise, denominados passos de análise ou simplesmente passos.

Desta forma, em cada passo de análise irá ocorrer o disparo simultâneo de todas as transições habilitadas e prontas (como em secção posterior será detalhado, poderá ser necessário recorrer ao conceito de micropassos de análise, correspondentes à capacidade de ordenar temporalmente eventos simultâneos).

Para garantir coerência com o paradigma dos sistemas reactivos, admite-se que as mudanças ocorridas no mundo exterior ao sistema modelado se realizam a um ritmo mais lento do que o ritmo associado aos passos de análise considerados.

Desta forma, tudo se passa como se o tempo de execução da especificação seja nulo (dado que um passo é instantâneo).

As soluções consideradas para a implementação do nível conceptual exposto, recorrem à utilização de um ciclo de tratamento para a execução do modelo, como implementação do conceito de passo, ou seja de um relógio, no sentido normalmente utilizado nos sistemas sequenciais síncronos. Deste modo o instante de análise é “estendido”, passando a ter uma duração não instantânea. É uma atitude

comum em sistemas monoprocessador, nomeadamente em controladores lógicos programáveis onde o tempo de ciclo é um dos parâmetros a ajustar pelo programador do sistema em função do tempo de execução necessário para a aplicação em causa.

As tarefas a serem executadas durante um ciclo de tratamento dividem-se em três grupos. O primeiro grupo é responsável pelo processamento das entradas e o terceiro grupo pelo das saídas, enquanto o segundo grupo corresponde à execução do modelo propriamente dita.

O grupo responsável pelo processamento das entradas é decomposto em duas tarefas:

- tarefa (a): aquisição dos sinais de entrada;
- tarefa (b): processamento de baixo-nível dos sinais de entrada, permitindo obter as variáveis associadas, através dos processos de codificação referidos, nomeadamente mapeamento, segmentação e codificação difusa (“fuzzification”).

O grupo responsável pelo processamento do modelo propriamente dito é decomposto em duas tarefas:

- tarefa (c): avaliação de eventos e condições em relação à componente não-autónoma;
- tarefa (d): determinação do estado seguinte.

O grupo responsável pelo processamento das saídas é decomposto em três tarefas:

- tarefa (e): determinação dos valores das variáveis de saída, através de processos de descodificação, nomeadamente mapeamento e descodificação difusa (“defuzzification”);
- tarefa (f): actualização dos sinais locais e remotos;
- tarefa (g): actualização dos sinais externos, através de redes de comunicação.

Em situações em que o tempo de ciclo necessário para a aplicação em causa seja muito maior que os tempos de processamento do conjunto das tarefas referidas, a execução das tarefas poderá ser realizada de modo sequencial, seguido de um tempo de letargia até ao final do ciclo (é, a título de exemplo, o modo de funcionamento adoptado nos controladores lógicos programáveis industriais).

Em situações em que os tempos de ciclo e de processamento sejam próximos, pode ser justificável o entrelaçamento do processamento das tarefas relativas a ciclos de tratamento consecutivos, como apresentado na parte superior da Figura 2-9, de modo a garantir condições de estabilidade em relação aos sinais de entrada.

Importa enfatizar que a técnica de implementação referida é apenas uma das possíveis, particularmente adequada às plataformas de execução que se pretendem considerar; corresponde, assumidamente, a um desvio em relação à atitude de execução instantânea do modelo, passo a passo, inicialmente proposta. Considerando, no entanto, válidos os condicionantes referidos, das taxas de variação das entradas em relação aos tempos de reacção do modelo, é um desvio tolerável.

Dado que, como referido no capítulo anterior, é de interesse considerar a execução paralela de várias especificações, eventualmente associadas à tradução de modelos especificados com recurso a vários formalismos, importa, ainda, considerar a possibilidade dos tempos de execução associados às várias tarefas paralelas serem muito diferentes; nessas situações e tendo em conta os condicionalismos específicos da aplicação, devem ser considerados ambientes de execução com diferentes definições de tempo, como apresentado na Figura 2-9 em que o tempo de ciclo associado à execução do modelo M1 é de uma unidade, enquanto que para a execução do modelo M2 é de quatro unidades.

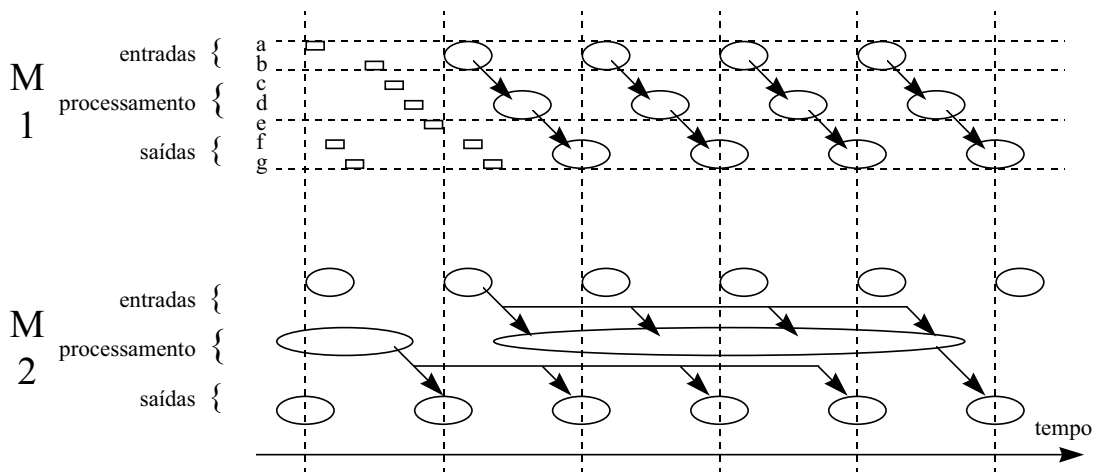


Figura 2-9 - Execução recorrendo a várias definições temporais,

indicando-se os tempos de execução das várias tarefas, individualmente e por grupos.

Conceptualmente, a modelação do processo de execução de especificações com diferentes ritmos pode ser considerada como a execução de vários processos ao ritmo mais elevado, em que os processos associados às especificações de granularidade temporal não unitária se obtêm considerando dois níveis de actividade hierarquicamente dependentes, sendo o nível superior constituído por uma máquina de estados que controla a execução das várias tarefas durante os vários ciclos.

### 2.2.7 Introduzindo dependências temporais

A modelação de dependências temporais nas RdP-R pode ser realizada de dois modos:

- considerando a existência de um relógio externo que interage com o modelo através da sua componente não-autónoma explícita, isto é, através de avaliação de eventos e condições, externos às características autónomas, onde se avalia o estado do relógio;
- através de um atributo de cor especial, modelando uma duração e denominado abreviadamente por “ttl” (de “time to leave”).

O primeiro modo referido decorre da normal execução do modelo especificado, pelo que não carece de detalhe adicional.

O segundo método proposto traduz a atitude geral de modelar o tempo associando-o aos lugares, ou, mais especificamente neste caso, dado se utilizar uma RdP Colorida, às marcas.

Como qualquer característica de cor, o atributo “ttl” pode ou não ser explicitamente referido nas inscrições de arco, bem como nas guardas das transições.

Caso não seja referido nessas expressões, assume-se uma semântica de omissão que exclue dos possíveis vínculos verificáveis a analisar todas as marcas com atributo “ttl” diferente de zero, ou seja, só as marcas sem atributo “ttl” ou com o atributo “ttl” igual a zero são utilizáveis para o disparo das transições.

Caso o atributo “ttl” seja referido nas inscrições de arco ou nas guardas, deve ser utilizado como qualquer outro atributo de cor, permitindo a modelação de situações de exceção (como, por exemplo, “time-outs”).

A execução do modelo, implicitamente, decrementa, até zero, o valor do atributo “ttl”, no início de cada passo de análise.

A Figura 2-10 apresenta várias situações típicas:

- os modelos representados em a), b) e c) são comportamentalmente equivalentes, impondo um atraso de  $N=3$  passos entre o disparo da transição  $t_1$  e da transição  $t_n$ ; é óbvia a vantagem em termos de simplificação do modelo resultante;
- o modelo representado em d) permite a evolução do processamento a partir da marcação do lugar  $p_1$  numa de duas direcções: a primeira, representando a evolução normal do modelo após sincronização com a marcação do lugar  $p_2$ , correspondendo ao disparo da transição  $t_1$ ; a segunda, modelando a situação de tempo de espera excedido (“time-out”), correspondendo ao disparo da transição  $t_2$  (a expressão do arco  $p_1 \rightarrow t_2$  ao não utilizar explicitamente o atributo  $ttl$ , só permitirá transportar marcas cujo atributo seja zero).

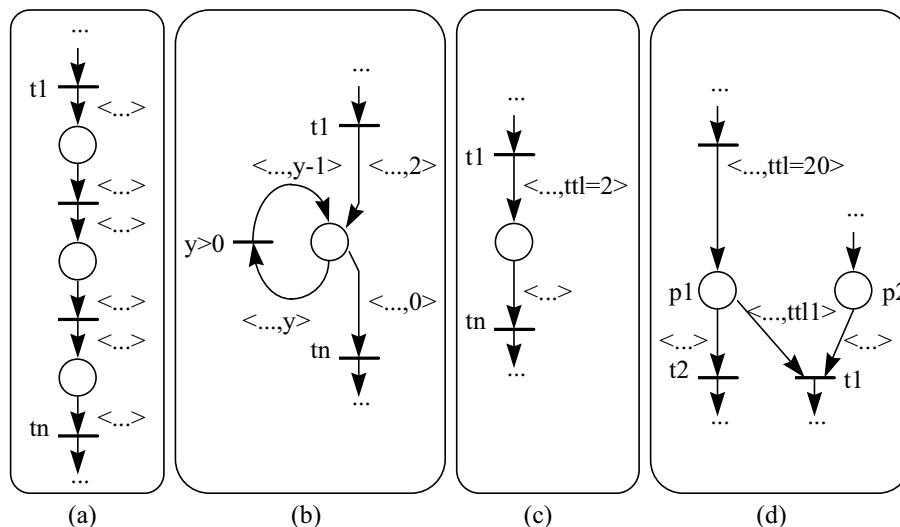


Figura 2-10 - Utilização do atributo de duração “ttl”.

Conceptualmente, o modelo comportamentalmente equivalente ao modelo proposto pode ser definido considerando que a todos os lugares da RdP se associam ciclos constituídos por uma transição prioritária, que tem associado o evento de início de passo e que provoca a transformação do atributo de cor “ttl” em todas as marcas existentes, decrementando-o de uma unidade, caso seja inicialmente maior que zero, a exemplo do utilizado na Figura 2-10(b).

Desta forma, a modelação da dependência temporal nas RdP-R é tratada de forma semelhante a qualquer outra característica de cor do modelo autónomo. A dependência temporal é, assim, transferida (ou escondida) para o nível da execução do modelo, através da definição do tempo de ciclo que estará associado a uma unidade dos “ttl” especificados.

Importa referir que a solução proposta, apresentada inicialmente em [Gomes & Steiger-Garção, 92], tem, relativamente à proposta apresentada em [Jensen, 95] para as RdP Coloridas Temporizadas, a vantagem significativa de permitir obter um espaço de estados finito, representativo da execução do modelo, isto é, que é comportamentalmente equivalente ao modelo especificado, podendo ser utilizado para a sua análise, simulação ou realização.

Nas RdP Coloridas Temporizadas, dado que se considera o tempo e a especificação de dependências temporais em termos do número de ordem do passo de análise, a ocorrência de um mesmo estado de marcação (em termos das características comuns das marcas) em dois instantes de análise, corresponde a dois estados do sistema (dado a representação do tempo fazer parte da representação do estado do sistema). Para permitir a obtenção de uma representação finita do espaço de estados (designado nas RdP Coloridas por grafo de ocorrências) excluem-se as referências ao atributo de tempo associado às marcas, o que restringe a sua utilização à análise do modelo, excluindo-se, portanto, as capacidades de simulação e execução com base no grafo de ocorrências.

Na construção do grafo de estados associado a um modelo RdP-R, os estados são representados exaustivamente (em extensão), como se apresentará no capítulo seguinte. Deste modo, as RdP-R, tendo a vantagem de se poder utilizar o grafo de estados para a verificação e implementação do modelo, têm a desvantagem de poderem dar origem a fenómenos de explosão do número de estados.

### 2.2.8 Modelação de fluxos de controlo e de dados

Uma parte significativa das aplicações das RdP à modelação de sistemas endereça as questões da especificação do controlo; é esse o caso nas aplicações comuns a sistemas de automação, de produção e a sistemas digitais. Em menor número são aquelas que utilizam as RdP para modelar os dados processados. As RdP Coloridas têm vindo a ser extensivamente utilizadas nesse sentido; como exemplo de especial interesse, refiram-se as aplicações à modelação de protocolos de comunicação [Billington, 95], onde os dois aspectos (controlo e dados) são tratados de forma conjunta.

No presente trabalho, a atitude de interpretação de um fluxo de dados e de um fluxo de controlo tem como referência os sistemas digitais, nomeadamente as arquitecturas microcontroladas, onde a um fluxo de dados está, normalmente, associado um caminho reactivo com comportamento assíncrono e a um fluxo de controlo um caminho sequencial contendo vários passos.

A modelação conjunta destes dois tipos de evolução é conseguida nas RdP-Reactivas através da utilização de dois tipos de transições, com disciplinas de disparo específicas. As condições de habilitação e prontidão são determinadas de forma semelhante para ambos os tipos de transição de acordo com o exposto, sendo o disparo efectivo da transição realizado de acordo com o seguinte:

- para as transições de tipo I ou assíncronas, apenas as marcas presentes no início do passo de análise são utilizadas para o cálculo dos vínculos possíveis; são representadas por uma barra, como normalmente;
- para as transições de tipo II ou síncronas, as marcas geradas no próprio passo de análise podem ser utilizadas nos seus vínculos possíveis; são representadas por uma barra, à qual se adicionam dois traços nas extremidades.

No sentido de ilustrar a utilização dos dois tipos de transições, considere-se a RdP-R elementar da Figura 2-11. A transição do estado B para o estado C é possível através da ocorrência de b. No entanto, a partir de C, caso ocorra unicamente c, evolui-se para o estado A, mas caso ocorram a e c, retorna-se ao estado B, transitando instantaneamente pelo estado A.

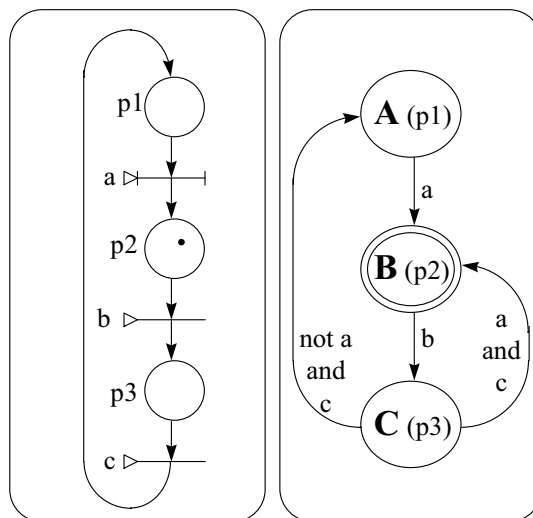


Figura 2-11 - RdP-R elementar e espaço de estados associado.

Desta forma, é possível que durante um passo de análise se efectuem disparos de transições em sequência. Cada estado intermédio dessa sequência é atingido durante um ou vários micropassos. Desta forma, um passo de análise pode ser subdividido num conjunto de micropassos. Os estados que são visitados instantaneamente são denominados evanescentes.

Do ponto de vista macroscópico, consideram-se apenas os estados anterior e posterior ao passo de análise, bem como o disparo simultâneo de todas as transições constituintes da sequência (como se de evoluções paralelas se tratasse).

Importa realçar que a adopção da disciplina de disparo síncrono pode, potencialmente, conduzir a indeterminismo. No caso do exemplo apresentado na Figura 2-11, considerando que as três transições presentes seriam síncronas e que os eventos a, b e c ocorrem simultaneamente, não é possível determinar o estado seguinte ao passo de análise.

As situações de indeterminismo não são permitidas, considerando-se o modelo inválido, e são detectadas pela verificação da ocorrência de um ciclo na sequência de estados evanescentes, isto é, quando na sequência de micropassos se encontra um estado já visitado durante essa sequência.

Importa realçar que as duas semânticas de disparo podem ser integradas numa única, suportada por um ambiente de execução baseado na definição de passo de análise e na utilização de um novo atributo de cor, booleano, denominado “existência”, que armazena a informação se a marca foi criada no passo actual ou num passo anterior. Todas as marcas geradas no passo em análise são criadas com o atributo “existência” igual a “actual”. No início de cada passo, os atributos “existência” de todas as marcas passam a armazenar o valor “anterior”. Na avaliação da habilitação das transições assíncronas só são utilizadas marcas com o atributo “existência” igual a “anterior”, enquanto que para as transições síncronas podem ser utilizadas marcas com um valor do atributo “existência” qualquer.

### 2.2.9 Modelação de saídas

A modelação das saídas do sistema é realizada através dos mecanismos normalmente utilizados na generalidade das classes com aplicações à modelação de controlo (a exemplo do realizado para as entradas); ver, por exemplo, as contribuições dos trabalhos de René David ou Manuel Silva.

A actuação dos sinais de saída pode ser realizada através de um de dois métodos (propostos para as RdP Interpretadas em [Silva, 85]):

- o primeiro método associa a actuação ao estado de marcação da rede, isto é, aos lugares; corresponde a uma atitude semelhante à utilizada nas máquinas de Moore;
- o segundo método adopta a atitude das máquinas de Mealy, permitindo gerar eventos de saída aquando do disparo de transições.

Naturalmente, o primeiro método permite a modelação de saídas estáveis associadas a um estado da rede, enquanto o segundo permite a modelação de eventos de saída.

A utilização de condições de teste que permitam “filtrar” a actuação das saídas é uma característica de primordial importância. A forma proposta utiliza expressões com uma sintaxe do tipo “IF *condição* THEN *acção*”, em que *condição* representa uma expressão envolvendo características da marcação actual, bem como referências a variáveis de entrada, e *acção* representa a afectação pretendida da variável de saída.

Nos casos em que não se utilizem condições de teste, a expressão fica reduzida à indicação da acção pretendida. Nesta situação, quando a acção é associada a uma transição, será activada sempre que a transição seja disparada; quando a acção é associada a um lugar, será activada sempre que o lugar se encontre marcado.

Do ponto de vista conceptual, a utilização de uma saída condicional numa transição é equivalente à utilização de duas transições, a primeira produzindo a acção especificada e a segunda não a produzindo. Considere-se, para o efeito, a situação de uma transição  $t1$ , com uma condição externa associada,  $a$ , e uma saída  $c$  especificada através de  $IF\ b\ THEN\ c$ . A transição  $t1$  será desdobrada em duas transições,  $t1a$  e  $t1b$ , iguais do ponto de vista dos arcos de ligação e das características coloridas. Na primeira transição,  $t1a$ , que produz a acção  $c$  quando ocorre, a condição externa a utilizar será alterada para  $a*b$ ; na segunda transição,  $t1b$ , que não produz a acção  $c$  quando ocorre, a condição externa a utilizar será alterada para  $a*/b$ ;

Importa referir que um sinal de saída pode resultar da agregação de um conjunto de variáveis de saída. É esse necessariamente o caso quando se tratam sinais difusos, onde se procede à descodificação difusa (ou “defuzzificação”), mas também pode ocorrer em relação a saídas convencionais permitindo uma simplificação da especificação. Neste caso, a expressão responsável pela agregação das várias variáveis necessita de ser explicitamente indicada. Fica, deste modo, claro que a actuação de saídas, embora dependente da marcação dos lugares ou do disparo das transições, pode ser especificada de forma separada em relação ao modelo RdP-R.

Uma característica à qual neste trabalho se tem vindo a dar ênfase particular (e que será continuada) prende-se com a capacidade de utilização de uma notação gráfica distintiva em relação às várias características do modelo. Foi assim para representar os dois tipos de transições, bem como as dependências em relação aos sinais de entrada; será assim quando, em capítulo posterior, se abordarem as questões da representação hierárquica. E assim é, também, em relação à representação das saídas.

Desta forma, do ponto de vista da notação gráfica utilizada, a activação de uma saída é representada através de uma seta “a sair” de um dos extremos da barra representativa da transição ou do lugar aos quais esteja associada. A Figura 2-8 atrás apresentada ilustra ambas as situações. No sentido de tornar a representação gráfica mais legível, evitando confusão com as restantes expressões (nomeadamente as guardas), nos casos de interesse, podem as variáveis de saída ser referidas dentro de um hexágono, como ilustrado com a saída  $f$  da Figura 2-8(a). Idêntico procedimento pode ser seguido em relação à indicação das condições associadas às entradas. A notação gráfica alternativa, também possível e comumente utilizada, recorre à indicação das saídas após uma barra que se associa ao identificador do nó; a Figura 2-8(b) recorre a essa notação.

### **2.2.10 Condições e eventos internos**

De um modo geral, os eventos produzidos pelo disparo de uma transição podem ser utilizados como eventos de entrada de uma outra transição. De modo semelhante, as activações de saídas devidas a um determinado estado de marcação, podem ser utilizadas como condições associadas a uma transição.

Acresce o facto da comunicação entre especificações paralelas, eventualmente resultantes da tradução separada de submodelos recorrendo a formalismos distintos, também ser suportada por variáveis de entrada e saída.

Estão, assim, reunidas as condições para definir a semântica associada à comunicação intramodelo através de eventos e condições.

Adopta-se o paradigma de comunicação instantânea, utilizado no Esterel e nos statecharts, através do qual as saídas produzidas durante um determinado passo de análise podem ser vistas como entradas durante esse mesmo passo de análise.

Como detalhadamente debatido nos trabalhos sobre statecharts (ver, por exemplo, [Harel, 87] [Beeck, 94]), a adopção deste tipo de reacção instantânea pode conduzir a não-determinismo na execução, traduzido na impossibilidade de determinar um estado seguinte estável, e na necessidade de detalhe adicional sobre o modo de considerar a activação dos eventos ao longo da sequência de reacção, nomeadamente ao longo dos vários micropassos de um mesmo passo de análise.

No que respeita aos tratamentos dos eventos, consideram-se as seguintes duas propriedades do modelo:

- satisfazer o princípio da acção-reacção, de modo que um evento produzido no micropasso N, não é considerado activo nos N-1 micropassos anteriores;
- satisfazer o princípio da persistência dos eventos, utilizado nos statecharts temporizados [Kesten & Pnueli, 92], traduzido no facto de um evento produzido no micropasso N se manter activo durante os micropassos subsequentes dentro do passo em análise (a semântica da maioria das variantes de statecharts considera um evento activo unicamente no micropasso seguinte ao ser produzido [Harel & Naamad, 96]).

As situações de indeterminismo, que não são permitidas, considerando-se o modelo inválido, são verificadas através da identificação de uma segunda ocorrência de um estado evanescente dentro da sequência de micropassos em análise no presente passo.

No sentido de especificar completamente a semântica associada ao disparo de transições síncronas e assíncronas (como atrás definidas), importa deixar claro que, nos casos em que à transição esteja associada uma condição dependente de sinais gerados pela marcação da RdP-R, aplicam-se as mesmas condicionantes válidas para a transição.

Na condição associada a uma transição, podem existir três tipos de dependências:

- em relação a sinais e eventos externos;
- em relação a eventos gerados internamente pelo disparo de transições;
- em relação a sinais gerados internamente por marcações específicas.

Na avaliação da condição associada a uma transição assíncrona, consideram-se os sinais e eventos externos (que se mantêm durante o passo de análise), os eventos e os sinais gerados internamente por marcações existentes no início do passo.

Na avaliação de condições associadas às transições síncronas, para além do indicado no parágrafo anterior, adicionam-se os sinais gerados internamente durante os micropassos do passo em análise.

Importa referir que a semântica proposta é uma entre as possíveis semânticas adoptáveis.

### **2.2.11 Interdependência entre as partes autónoma e não-autónoma**

A possibilidade das características das RdP-R associadas às componentes autónoma e não-autónoma se poderem condicionar mutuamente, permite obter expressividade e níveis de compactação mais elevados, consequentes com a adopção da modelação de sinais externos.

Os condicionantes podem ser realizados nos dois sentidos, nomeadamente:

- embebendo sinais de entrada nas características coloridas da rede, através da sua referência nas inscrições de arcos e guardas; isto significa que os sinais de entrada podem determinar as características de marcas coloridas;
- determinando os eventos de entrada a utilizar numa transição a partir dos vínculos actuais possíveis da transição; como exemplo, refira-se a selecção de um sinal de entrada específico dentro de um vector de sinais de entrada, dependendo das características coloridas das marcas de um determinado vínculo possível.

Importa referir que as capacidades de modelação disponíveis nas RdP-R com a interdependência de características referida são semelhantes às capacidades do modelo que considera de forma separada as duas componentes, desde que os números de valores possíveis e de sinais de entrada utilizados sejam finitos.

Dito de outro modo, é possível encontrar o modelo “de componentes separadas” comportamentalmente equivalente a um modelo “de componentes interactuantes”, desde que os números de sinais de entrada e de valores possíveis desses sinais sejam finitos.

A validação desta afirmação pode ser comprovada, considerando exemplos elementares endereçando as duas situações básicas de dependência.

Na Figura 2-12(a) apresenta-se uma situação de selecção de um sinal de um vector de sinais a partir de um atributo da característica colorida, isto é, determinam-se características não-autónomas a partir

das características autónomas. Considerando que o vector de sinais de entrada  $a$ , possui dois elementos, o modelo comportamentalmente equivalente, apresentado na Figura 2-12(b), é obtido através da duplicação da transição envolvida; as novas transições são interligadas ao modelo de modo semelhante à transição inicial, respeitando arcos e respectivas inscrições; o evento associado a cada uma das novas transições corresponde a cada instância possível da expressão com eventos associada à transição inicial; a guarda associada a cada uma das novas transições é obtida adicionando-se à guarda inicial a restrição imposta à expressão com eventos. Um caso típico de aplicação da situação ilustrada, ocorre quando se considera um sistema composto por vários andares, semelhantes entre si, onde a variável  $i$  representa a ordem do andar. No caso geral, a replicação da transição inicial deverá ser igual ao número de instâncias possíveis da expressão com eventos associada à transição.

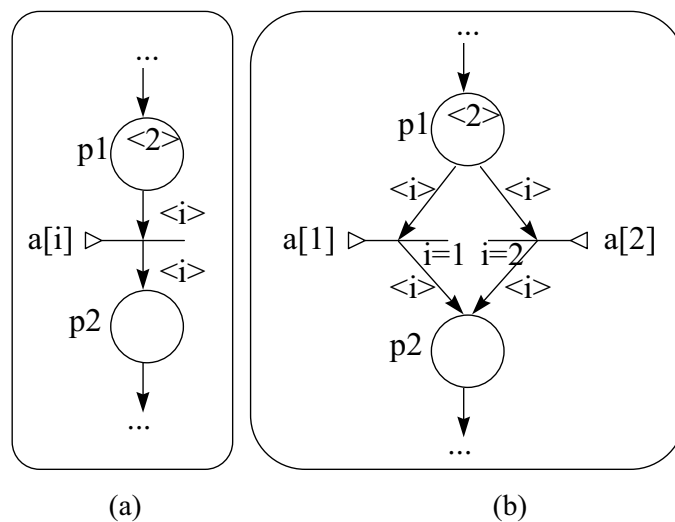


Figura 2-12 - Selecção de eventos através de atributos de marca.

Na Figura 2-13(a) modela-se uma situação de determinação de atributos de marcas geradas por uma transição, tendo em conta condições associadas a eventos externos. Isto é, uma situação em que se determinam características autónomas a partir de características não-autónomas. Considere-se uma situação em que se pretende realizar a sincronização entre a evolução de dois modelos em execução paralela, em que um dos modelos deverá passar ao outro um parâmetro; a Figura 2-13(a) modela essa situação em que a transição modela a sincronização entre uma condição local (marcação de p1) e uma condição remota (evento *received*); o parâmetro de passagem é considerado associado a esse evento e considera-se, por simplificação, que pode assumir dois valores, nomeadamente 'a' e 'b'. O modelo comportamentalmente equivalente, apresentado na Figura 2-13(b), é obtido através da duplicação da transição envolvida; as novas transições são interligadas ao modelo de modo semelhante à transição inicial, respeitando arcos e respectivas inscrições; o evento associado a cada uma das novas transições corresponde a cada instância possível da expressão com eventos associada à transição inicial; a guarda associada a cada uma das novas transições é obtida adicionando-se à guarda inicial a restrição

imposta à expressão com eventos. Um caso típico de aplicação da situação ilustrada será a de modelação das técnicas de codificação difusa a abordar em capítulo posterior. No caso geral, a replicação da transição inicial deverá ser igual ao número de instâncias possíveis da expressão com eventos associada à transição.

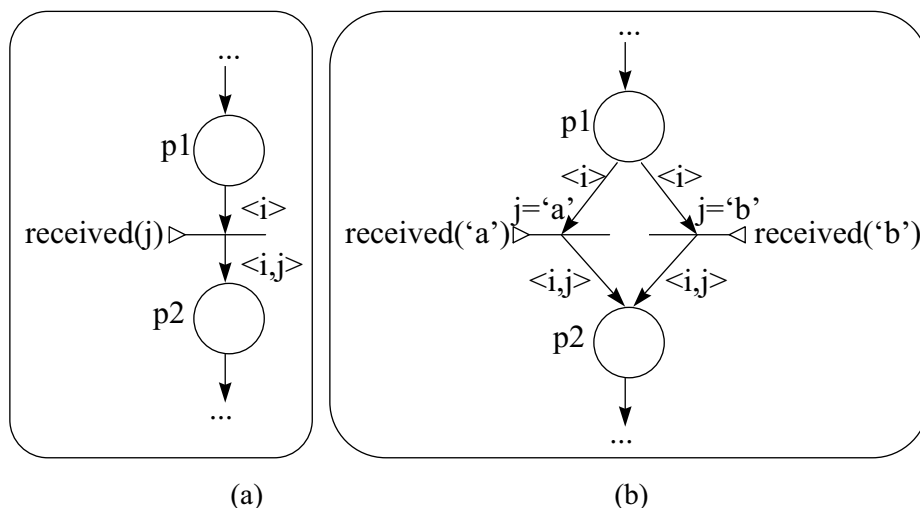
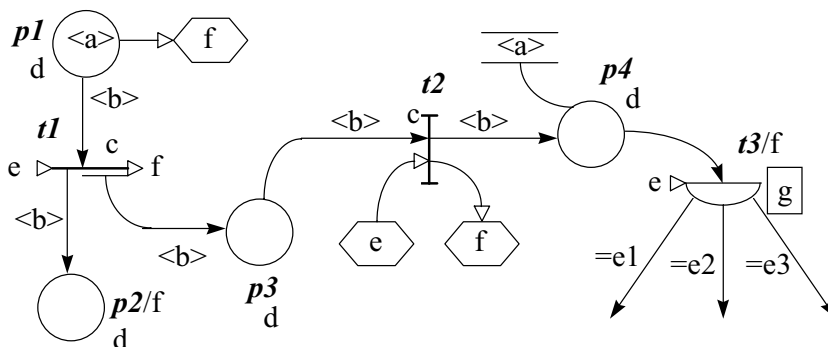


Figura 2-13 - Geração de marcas considerando variáveis externas associadas a eventos.

**2.2.12 Sobre a notação gráfica utilizada**



- a - marcação inicial;      b - expressão de arco;      c - guarda;      d - cor;
- e - eventos e condições;      f - acções;      g - prioridade.

Figura 2-14 - Notação gráfica proposta: exemplos.

No sentido de resumir as várias propostas apresentadas relativamente à notação gráfica utilizada, apresenta-se na Figura 2-14 um modelo ilustrando as várias situações típicas nas RdP-R.

A notação gráfica utilizada representa as transições por barras e os lugares por elipses (ou circunferências).

A representação das características da componente autónoma (RdP Colorida) são as seguintes:

- a marcação inicial pode ser representada dentro (ver lugar  $p1$ ) ou próximo do lugar respectivo, ou dentro de um par de linhas ligado ao lugar através de um arco (ver lugar  $p4$ );
- as expressões de arco são representadas próximas dos arcos respectivos; na Figura 2-14, utiliza-se a letra  $b$  para a sua representação;
- as guardas são representadas próximas das transições respectivas; na Figura 2-14, utiliza-se a letra  $c$  para a sua representação;
- a cor de cada lugar é representada próxima do lugar respectivo; na Figura 2-14, utiliza-se a letra  $d$  para a sua representação.

A representação das características da componente não-autónoma são as seguintes:

- os eventos e condições externos podem ser representados isoladamente ou dentro de um hexágono; ligam-se à transição respectiva através de uma seta (só contornada), com ou sem arco; na Figura 2-14, utiliza-se a letra  $e$  para a sua representação; as três transições utilizadas referem três especificações válidas;
- as acções podem ser representadas de diversas formas (na Figura 2-14, utiliza-se a letra  $f$  para a sua representação):
  - associadas ao identificador do nó (ver lugar  $p2$  e transição  $t3$ );
  - inseridas num hexágono e ligando-se à transição ou lugar respectivo através de um arco com seta (só contornada) (ver lugar  $p1$  e transição  $t2$ );
  - ligando-se à transição respectiva através de uma seta (só contornada) (ver transição  $t1$ );
- as transições comuns (denominadas assíncronas) são representadas por uma barra, enquanto que as transições síncronas possuem dois traços nas extremidades (ver trans.  $t2$ );
- as prioridades são representadas dentro de um rectângulo junto à transição (ver transição  $t3$ ); na Figura 2-14, utiliza-se a letra  $g$  para a sua representação;
- é possível representar de modo compacto fluxos de controlo dependentes de sinais externos, tipo estruturas IF-THEN-ELSE como ilustrado na transição  $t1$  e estruturas CASE como ilustrado na transição  $t3$ .

## 2.3 Análise de um exemplo

Dado que a simples enumeração das características das RdP-R (mesmo quando apresentada de forma justificada), não é suficiente para produzir uma visão completamente esclarecida sobre as suas capacidades, optou-se por apresentar nesta secção um exemplo simplificado que permita clarificar o posicionamento das RdP-R face a outros formalismos, em particular em relação aos três formalismos de referência (RdP Coloridas [Jensen, 92], RdP Interpretadas [Silva, 85] e statecharts [Harel, 87]). Pretende-se, também, ilustrar a flexibilidade de modelação de que o projectista dispõe, necessitando de seleccionar, de entre as características disponíveis, as que resultam num modelo mais expressivo.

Escolheu-se um exemplo simples que se possa comportar como um exemplo didáctico, extraído dos trabalhos realizados na área dos edificios inteligentes, tema a apresentar detalhadamente em capítulo posterior. Por razões de pedagogia, inicia-se pela caracterização do caso elementar, ao qual serão progressivamente adicionados novos requisitos.

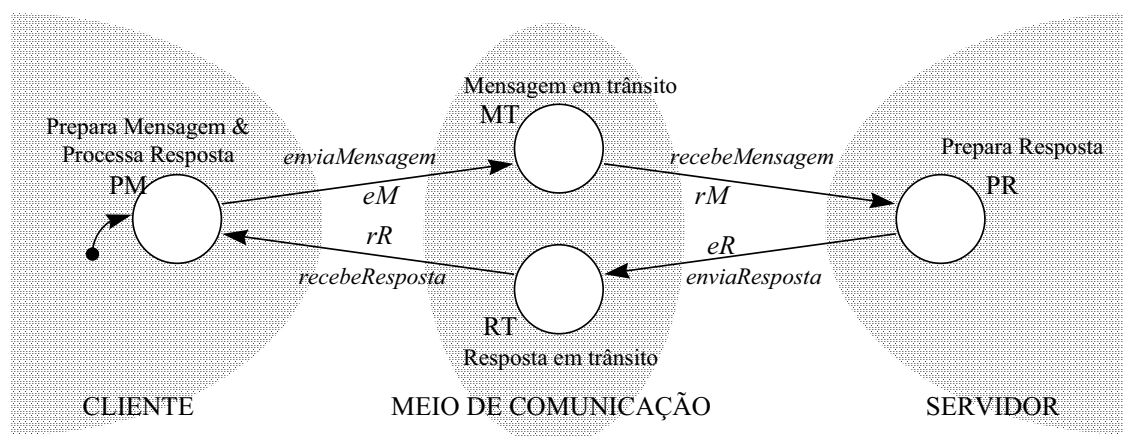


Figura 2-15 - Diagrama de estados modelando a comunicação de um cliente com um servidor.

Considere-se, pois, um sistema constituído por dois processos computacionais, interligados de acordo com o modelo cliente-servidor. O cliente (tipicamente um processo responsável pelo interface com o utilizador) tem como função principal apresentar o estado de um sensor, que é monitorado por um processo servidor. O servidor fornece o estado do sensor como resposta à recepção de uma mensagem com esse pedido, o fluxo de mensagens entre cliente e servidor pode ser modelado através da máquina de estados da Figura 2-15.

Como requisito seguinte, considere-se que o cliente é responsável pela actualização periódica de um sinóptico em que se encontram representados os estados de vários sensores. Para isso, necessita de contactar um conjunto de servidores no sentido de recolher essa informação (através da emissão

simultânea de mensagens para todos eles); quando receber a informação de todos os servidores procede à actualização do sinóptico. Para simplificar, consideram-se apenas dois servidores,  $A$  e  $B$ .

A tentativa de modelar o sistema descrito através de diagramas de estados revela uma das suas grandes limitações (identificada no início do presente capítulo), isto é, inabilidade para modelação de processos paralelos: o número de estados resultante da actividade independente de dois processos é igual ao produto do número de estados de cada uma das máquinas de estado.

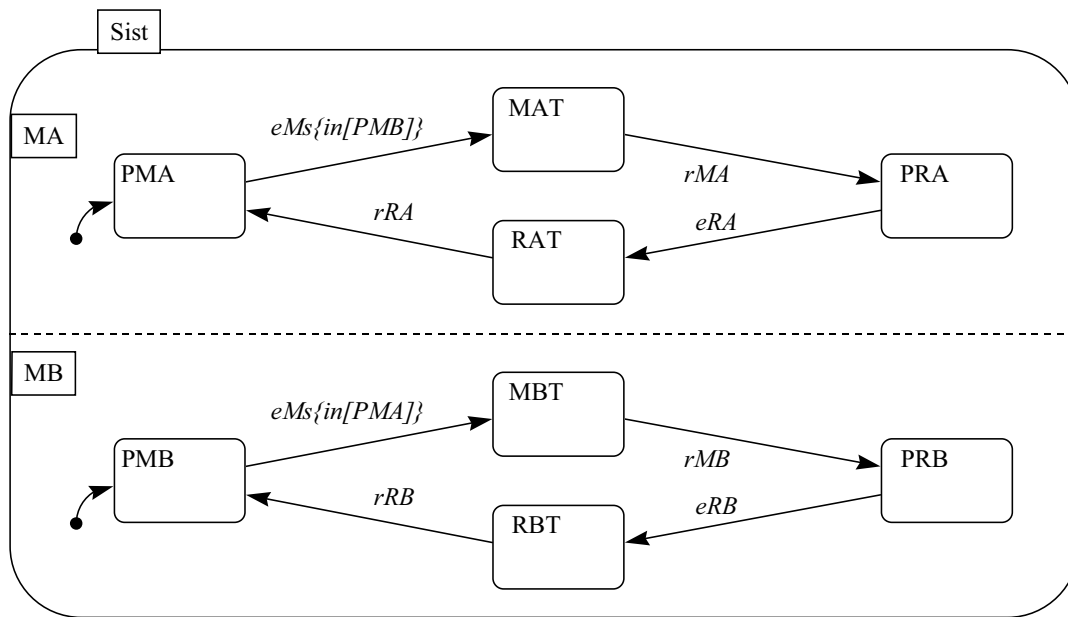


Figura 2-16 - Statechart modelando a comunicação de um cliente com dois servidores.

O statechart da Figura 2-16 permite modelar de forma compacta e expressiva o sistema descrito, garantindo que o crescimento do número de estados é linear. As referências utilizadas no statechart, quer no tocante a identificadores de estados, quer de eventos, mantêm-se coerentes com os utilizados no diagrama de estados prévio, tendo em conta o facto de se utilizarem duas máquinas de estado  $MA$  e  $MB$  para contactar com os dois servidores  $A$  e  $B$ .

A sincronização do envio das mensagens para os servidores é garantida através do condicionamento mútuo da evolução das duas máquinas de estado  $MA$  e  $MB$ , traduzido nas anotações presentes nos arcos de saída dos estados  $PMA$  e  $PMB$ . No caso da máquina de estados  $MA$ , só é permitido transitar do estado  $PMA$ , caso na máquina de estados ortogonal  $MB$  o estado actual seja  $PMB$  (garantido através da anotação  $in[PMB]$ ). Idêntica restrição está presente na máquina de estados  $MB$ .

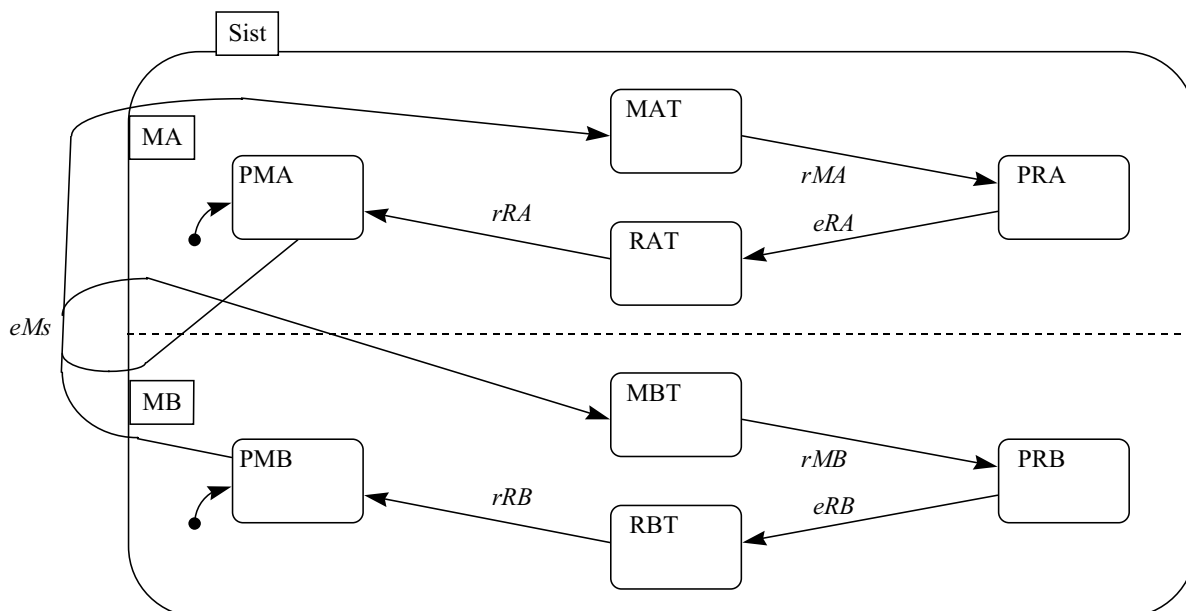


Figura 2-17 - Statechart equivalente ao apresentado na Figura 2-16.

Importa referir que os condicionamentos entre máquinas ortogonais realizados através de anotações podem, nalguns casos, ser explicitamente modeladas pela estrutura do statechart. É esse o caso do exemplo em análise, dando origem ao statechart da Figura 2-17, em que as restrições à evolução do statechart são tornadas explícitas através do arco composto dependente do evento  $eMs$ .

Qualquer dos modelos apresentados até aqui nesta secção podem, facilmente, ser traduzidos em RdP-R. Para isso, bastará substituir todos os arcos entre estados, por arcos contendo transições com dependências associadas nos mesmos eventos.

Em particular, na Figura 2-18 apresenta-se um modelo RdP-R (utilizando características das RdP Coloridas e RdP Interpretadas), resultado de uma tradução do statechart da Figura 2-17. As marcas coloridas utilizadas ( $\langle 1 \rangle$  e  $\langle 2 \rangle$ ) representam a comunicação com os dois servidores,  $A$  e  $B$ , respectivamente.

Finalmente, considerando as características coloridas do modelo, por um lado, associadas a características específicas das RdP-R, por outro, é possível encontrar o modelo RdP-R da Figura 2-19, elegante e expressivo. De notar a sua semelhança estrutural com o diagrama de estados da Figura 2-15 de onde partimos para modelar o sistema elementar. Os eventos utilizados associados às transições são vectores, permitindo que a determinação dos vínculos em cada transição se faça considerando em simultâneo a marcação actual e a ocorrência de eventos, ou seja, utilizando simultaneamente características autónomas e não-autónomas do modelo RdP-R.

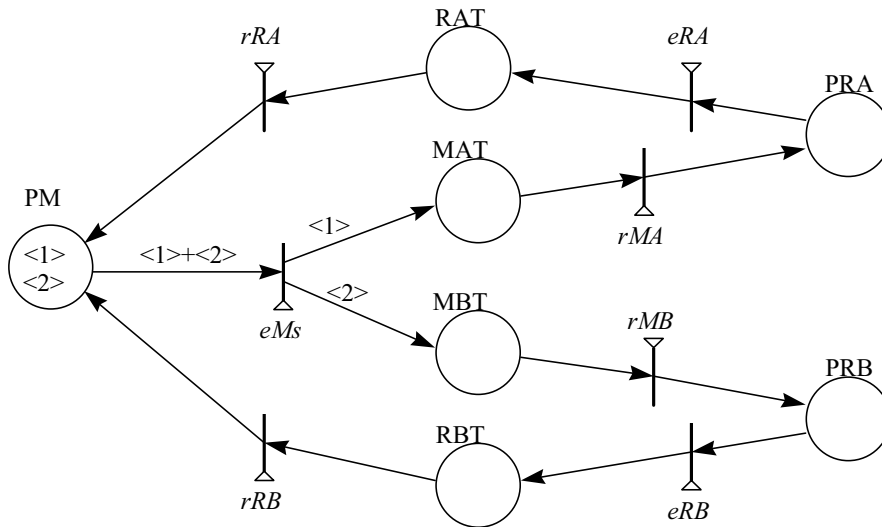


Figura 2-18 - RdP modelando a comunicação de um cliente e dois servidores.

De notar que a modelação de um sistema com  $N$  servidores, produziria pequenas alterações no modelo da Figura 2-19, nomeadamente na marcação inicial do lugar  $PM$ , que conteria  $N$  marcas, e das inscrições associadas aos arcos da transição de saída do mesmo lugar, que passariam a conter a

expressão  $\sum_{i=1}^N \langle i \rangle$ .

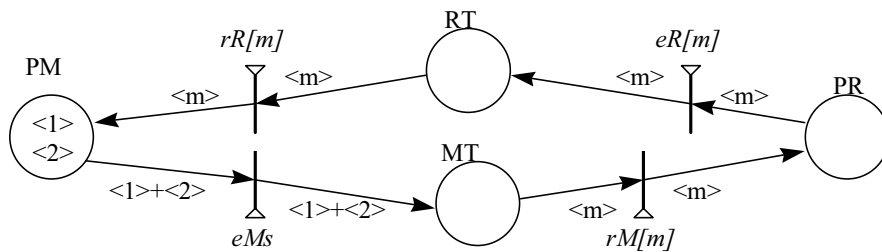


Figura 2-19 - RdP-R equivalente à RdP da Figura 2-18.

De modo semelhante ao afirmado para os statecharts apresentados, também será possível encontrar RdP-R comportamentalmente equivalentes deslocando anotações para estrutura e vice-versa. No particular das RdP Coloridas é uma opção há muito tempo conhecida e que, fundamentalmente, é determinada pela formação e prática de quem produz o modelo RdP. Como situações paradigmáticas refiram-se dois casos extremos:

- na primeira situação, em que, com o objectivo de se obter um modelo muito compacto, se desloca todo o esforço de modelação para as anotações, resultando um modelo “pequeno”, mas de difícil apreensão, devido à densidade das anotações coloridas da rede;

- na segunda situação, em que, com o objectivo de explicitar graficamente todas as dependências e aspectos do sistema, se é conduzido a uma rede de tal forma extensa, inviabilizadora de uma apreensão do seu comportamento.

No caso das RdP-R, para além da busca dos equilíbrios referidos no último parágrafo entre anotações coloridas e estrutura da rede, é necessário considerar uma procura de um equilíbrio de novo tipo, que tem a ver com as características não-autónomas do modelo, isto é, entre as dependências modeladas através de eventos e sinais e as modeladas através da estrutura da rede. Em suma, mais uma área em que a criatividade e as capacidades do projectista serão evidenciadas...

Tendo em conta o objectivo da apresentação do exemplo, não se considera necessário ilustrar outras vantagens resultantes da utilização das RdP-R, algumas das quais são vantagens bem conhecidas das RdP. De entre elas, refiram-se, a título de exemplo:

- as capacidades de modelação de dados (que permitiriam modelar os conteúdos das mensagens propriamente ditos),
- as capacidades de memorização (que permitiriam modelar os protocolos para recuperação de erros de comunicação e modelar memórias de comunicação, vulgo “buffers”),
- as características de modelação local (por oposição aos diagramas de estado, em que se necessita de conhecer o estado global do sistema) permitindo atitudes de modelação incremental e de refinamentos específicos a submodelos,
- as capacidades de modelação de temporizações (nomeadamente “*time-outs*”).



## 3. Aspectos de verificação e de implementação

### *Sumário do capítulo*

No presente capítulo serão abordados alguns temas relevantes ligados à análise e execução de modelos RdP-Reactivas.

Assim, o capítulo é composto pelas seguintes secções:

*3.1 Sobre as técnicas de verificação e execução de RdP-R*

*3.2 Sobre ambientes de desenvolvimento*

*3.3 Sobre a construção e utilização do grafo de estados*

*3.4 Resolução de conflitos*

Na primeira parte, as questões associadas a diferentes tipos de soluções para a execução das RdP-R serão apresentadas. Para além dos tradicionais jogadores-de-marcas, responsáveis pelo que é referido na literatura como o jogo das marcas (isto é, o resultado do disparo das transições da RdP), procurar-se-á dar ênfase à potencial utilização do espaço de estados associado ao modelo RdP-R, como base para a execução do modelo em plataformas de recursos computacionais limitados. Um segundo aspecto importante a abordar nesta secção é o das técnicas de análise aplicáveis às RdP-R, permitindo proceder à verificação do modelo.

A segunda secção apresenta, brevemente, os protótipos desenvolvidos para edição e simulação de RdP.

A construção do espaço de estados associado a um modelo RdP-R será debatida na terceira secção. Será, nomeadamente, apresentado o algoritmo para a sua construção e ilustrada a sua execução em controladores lógicos programáveis e dispositivos de lógica programável.

Na última parte abordar-se-ão questões associadas à resolução automática de conflitos, tema imprescindível para a geração automática de código para execução. Analisar-se-ão as várias atitudes possíveis na resolução automática de conflitos em RdP-R, considerando uma taxonomia baseada em três dicotomias “tradicionais”:

- sistemas determinísticos versus não-determinísticos,
- resolução de conflito equitativa versus não-equitativa, e
- execução baseada num jogador-de-marcas versus jogador-de-estados.

## 3. Aspectos de verificação e de implementação

### 3.1 Sobre as técnicas de verificação e execução de RdP-R

A execução das RdP-Reactivas necessita de considerar, de forma ponderada, factores potencialmente contraditórios. Por um lado, devido às suas características (enunciadas no capítulo anterior) e complexidade associada, necessitam de um suporte computacional sofisticado. Por outro lado, desde cedo neste trabalho se identificaram as plataformas de baixo custo, nomeadamente os controladores industriais e os dispositivos de lógica programável, como interessantes para executarem o modelo de controlo especificado. Deste modo, as técnicas a seleccionar para a execução das RdP-R são de particular interesse.

#### 3.1.1 Jogador-de-marcas versus jogador-de-estados

Do ponto de vista da execução das RdP-R, duas soluções básicas serão analisadas nesta subsecção: a primeira recorre à utilização de um jogador-de-marcas enquanto a segunda se baseia na execução do grafo de estados associado (encarado como “espaço transformado” da representação inicial). A Figura 3-1 apresenta as várias soluções consideradas.

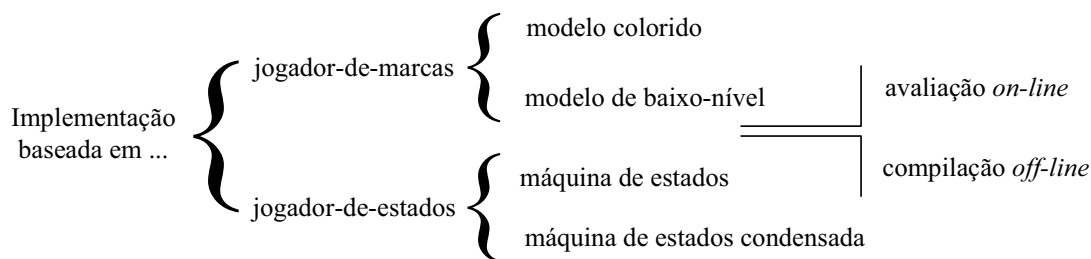


Figura 3-1 - Mapa de soluções para a execução das RdP-R.

A expressão jogador-de-estados é utilizada para referir um jogador-de-marcas simplificado para a execução de máquinas de estado, por analogia óbvia com a expressão jogador-de-marcas vulgarmente utilizada na literatura da especialidade.

O primeiro grupo de soluções, que recorrem a um jogador-de-marcas, correspondem à solução comumente utilizada. A plataforma computacional de execução deverá ser adequada à complexidade do modelo; no caso das RdP-R, computadores pessoais ou estações de trabalho (“*workstation*”) são consideradas como plataforma mínima para a sua execução. De um modo geral, duas soluções principais podem ser consideradas para um jogador-de-marcas de uma RdP-R:

- a primeira utiliza o modelo RdP-R propriamente dito,
- a segunda recorre à descompactação das características coloridas, no sentido de obter o modelo de baixo nível equivalente, isto é, eliminando as cores e anotações associadas. Esta solução não será alvo de desenvolvimentos posteriores neste trabalho.

Em qualquer dos casos, importa realçar a característica de que num jogador-de-marcas, o modelo é analisado em tempo de execução, isto é, *on-line*, impondo uma relação de proporcionalidade entre a dimensão do modelo a executar e os recursos computacionais da plataforma de execução.

O segundo grupo de soluções, que utilizam o espaço de estados e respectivo jogador-de-estados, corresponde a transferir a complexidade associada à execução das RdP-R e sua avaliação numa situação *on-line* para uma situação de avaliação *off-line*. De certa forma, corresponde à compilação da especificação. Duas soluções principais serão analisadas neste capítulo:

- a primeira utiliza a máquina de estados resultante da análise da especificação inicial,
- a segunda utiliza uma representação condensada (compactada) da máquina de estados obtida.

Este grupo de soluções, utilizando o espaço de estados, sofre de uma limitação muito significativa, associada ao fenómeno da explosão do espaço de estados de uma especificação modelando actividades paralelas independentes. Nessa situação, o espaço de estados global resulta do produto cartesiano dos espaços de estados concorrentes. Esta é uma das razões significativas para a sua (quase) não utilização como método de execução de RdP. No entanto, embora de aplicação limitada (em condições que serão identificadas nos parágrafos seguintes) revela-se um método de interesse.

A implementação baseada na execução do espaço de estados recorre à análise prévia do modelo RdP-R e construção *off-line* do grafo de estados associado. O espaço de estados associado a um modelo RdP é, na bibliografia da área, caracterizado de várias formas e, conseqüentemente, referido através de vários termos, nomeadamente árvores de acessibilidade ou alcançabilidade, ou grafos de ocorrências (na bibliografia das RdP Coloridas), de acessibilidade ou alcançabilidade. São, no

entanto, termos utilizados no âmbito da análise de RdP Autónomas, pelo que neste trabalho, se preferirão referências a “espaço de estados”, “grafo de estados” ou “máquina de estados equivalente”.

Nesta solução, a execução do modelo é realizada através da execução *on-line* do grafo de estados produzido (uma máquina de estados).

Importa referir que considerar o espaço de estados como suporte directo para a implementação é uma solução apenas aplicável quando o modelo não é  *muito* complexo, ou seja, o número de estados é *relativamente* pequeno.

Para um grande número de modelos relativos a aplicações de engenharia, o grafo associado ao espaço de estados pode ter um número *demasiado* elevado de estados, conduzindo à sua intractabilidade, do ponto de vista dos sistemas embebidos. Esta é uma das principais razões pela qual as soluções baseadas na execução do espaço de estados não são comumente utilizadas.

Deste modo, nas situações em que se pretenda utilizar controladores de baixo custo como plataformas de execução das RdP-R, estamos perante características contraditórias: por um lado os limitados recursos de computação da plataforma, por outro as necessidades de um jogador-de-marcas coloridas (em termos gerais, a implementação de jogadores-de-marcas coloridas em controladores de baixo custo não é possível). A utilização do grafo de estados e de um jogador-de-estados apresenta-se, assim, como uma solução de elevado interesse para estas situações.

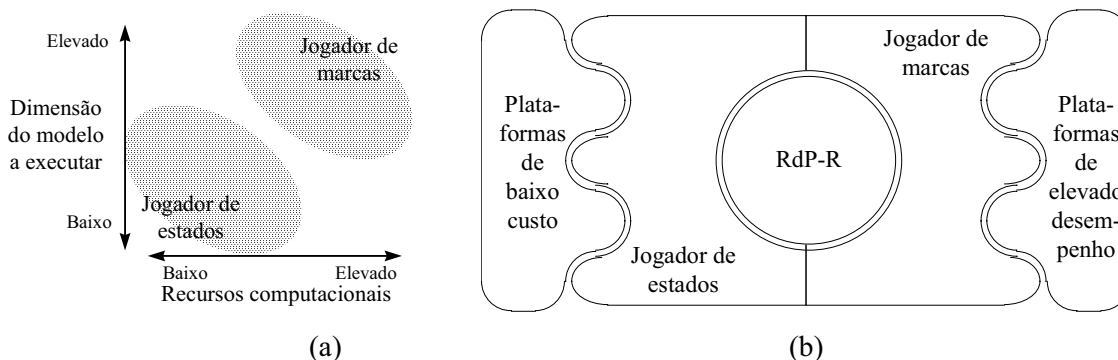


Figura 3-2 - A solução à medida: a) no plano “dimensão do modelo a executar versus recursos computacionais disponíveis”; b) adaptação das RdP-R a plataformas de execução específicas.

A Figura 3-2 evidencia a necessidade de utilizar jogadores-de-marcas ou jogadores-de-estados para a adequação das RdP-R a plataformas de execução com diferentes capacidades. Na Figura 3-2(a) evidenciam-se as áreas de aplicação dos jogadores-de-marcas e jogadores-de-estados em termos dos recursos computacionais disponíveis versus a dimensão do modelo a executar. A dimensão do modelo a executar considera-se proporcionalmente dependente da dimensão da especificação (em termos do

número de nós do grafo) e do nível de compacidade do modelo (desde as RdP Coloridas, que permitem índices elevados de compactação do modelo, aos diagramas de estados, correspondentes ao menor índice de representação compactada). Na Figura 3-2(b) evidencia-se o papel de adaptação das RdP-R a plataformas de execução específicas, nomeadamente os casos típicos de plataformas de baixo-custo e reduzido desempenho (controladores industriais e dispositivos de lógica programável), bem como a plataformas de elevado desempenho computacional (“workstations”).

No sentido de enfatizar o importante papel que as implementações baseadas na execução do espaço de estados podem assumir, em termos de flexibilidade e utilidade em aplicações específicas de automação, refira-se que:

- a arquitectura subjacente à execução de máquinas de estado é muito simples e implementada pela maioria de plataformas *hardware*, incluindo as de (muito-)baixo nível, isto é, de recursos muito limitados;
- a verificação e validação de uma especificação baseada numa máquina de estados dispõe, actualmente, de inúmeras ferramentas, algumas delas extensivamente testadas e integradas em ambientes de desenvolvimento de sistemas; aumenta-se, desta forma, a robustez da solução e a sua aceitação do ponto de vista dos meios industriais, permitindo, desta forma, embeber suportadamente modelos RdP em aplicações do mundo real;
- evoluções (alterações) nas características das RdP-R apenas terão impacto nas ferramentas para processamento *off-line* do modelo (compilação da especificação), permitindo manter inalterados o núcleo para execução em tempo-real bem como o conjunto de ferramentas de verificação e validação (baseadas na máquina de estados associada); deste modo, a robustez da implementação é mantida ao longo de eventuais alterações das características das RdP-R (as alterações às características de uma classe são, normalmente, realizadas no sentido de adequar as suas capacidades de modelação tendo em vista uma área de aplicação específica).

Por estas razões, as implementações baseadas na execução do espaço de estados através de jogadores-de-estados aparecem tão interessantes como os jogadores-de-marcas (com os domínios de aplicação ilustrados na Figura 3-2) e serão, neste trabalho, especialmente consideradas.

Para além dos argumentos expostos, a discussão sobre as vantagens e desvantagens da utilização do grafo de estados como suporte à execução pode encontrar um paralelo nos argumentos utilizados na discussão sobre execução de programas compilados e interpretados (embora nesse caso não se

coloquem as questões associadas à explosão do espaço de estados). Esta é, também, uma dicotomia de forte impacto na implementação das RdP-Reactivas.

### 3.1.2 Compilação versus interpretação

A linguagem seleccionada para expressar as anotações do modelo (inscrições nos arcos, guardas, eventos, acções, cores e marcas) necessita de ter em conta a solução considerada para a execução. Como exemplos significativos das duas abordagens (interpretação versus compilação), refiram-se:

- o ambiente de desenvolvimento para RdP Coloridas, Design/CPN, utilizando uma extensão à linguagem interpretada SML, permite a execução do modelo através de um jogador-de-marcas;
- as CpPNeTS [Barros, 96] que, recorrendo a uma representação das várias características da RdP em C++, permite a execução do modelo após compilação através de um compilador de C++; o espaço de estados associado é um dos resultados dos processamentos possíveis.

A atitude seguida neste trabalho (e noutros por ele enquadrados, como em [Barros, 96]) é a de optar pela compilação das especificações, como forma de utilizar adequadamente o código executável gerado, adaptando-o para diversas plataformas e garantindo a satisfação das restrições de tempo implícitas ao modelo. Embora tenha a desvantagem de tornar mais “pesado” o ambiente em que se procede à especificação, verificação e simulação do modelo, encontra-se, claramente, vocacionada para o suporte à implementação do modelo em sistemas autónomos; esta característica constitui uma enorme vantagem relativamente aos sistemas actualmente disponíveis.

Refira-se, a propósito e ilustrando claramente as vantagens da utilização de especificações compiladas, a experiência de aplicação de RdP Coloridas, com recurso ao Design/CPN, à modelação de um sistema de controlo de acessos e de intrusão em instalações de empresas [Rasmussen & Singh, 96], desenvolvido num trabalho de Mestrado na Universidade de Aarhus, Dinamarca: devido ao tempo de reacção do ambiente não ser compatível com os requisitos de tempo-real do sistema a controlar, o modelo RdP Colorida foi unicamente utilizado para verificação e validação do sistema, tendo a implementação final sido realizada em C++, por tradução não-automática da especificação em SML.

Com a disponibilização de um ambiente que permita obter código executável de forma automática a partir da especificação da rede de alto-nível, como o que se persegue neste trabalho, contribuir-se-á,

de forma significativa, para o colmatar dessa lacuna no conjunto de ferramentas actualmente disponíveis.

### 3.1.3 Da especificação à implementação

Considerando o processo de desenvolvimento de um sistema, onde os dois aspectos abordados nas subsecções anteriores se enquadram, importa identificar os métodos aplicáveis e/ou disponíveis para proceder à verificação e validação do modelo, bem como as atitudes utilizadas para a geração automática de código.

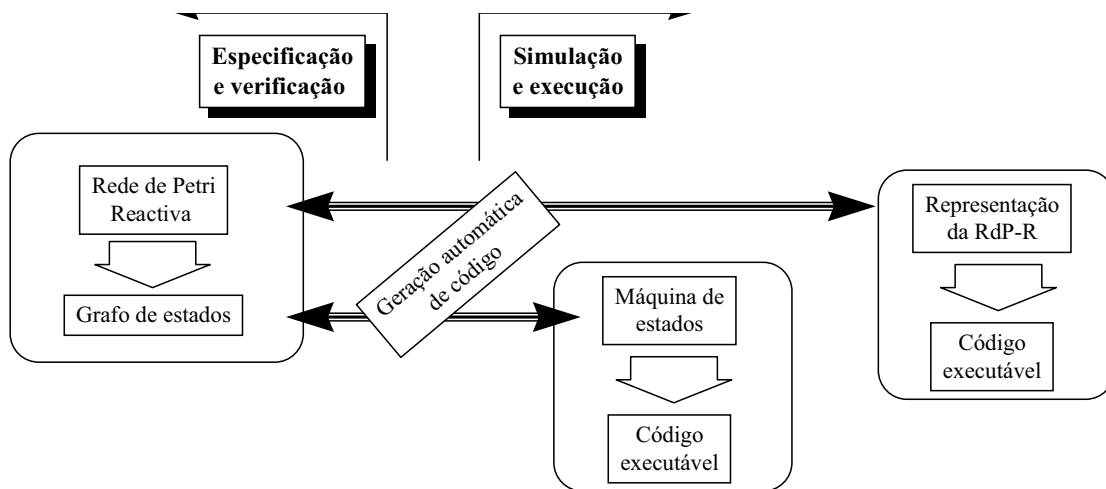


Figura 3-3 - Etapas do desenvolvimento.

Considere-se a Figura 3-3 como suporte para o que seguidamente se exporá. Nela se identificam duas áreas distintas:

- a primeira visando as actividades de especificação e verificação, isto é, análise de propriedades;
- a segunda orientada para a execução ou validação da especificação (através da simulação do modelo), com base na geração automática de código a partir dos resultados das fases de especificação e verificação.

Nas duas subsecções seguintes os dois aspectos serão abordados.

### 3.1.3.a Métodos de análise aplicáveis

No capítulo anterior foram apresentadas as características das RdP-R. Identificaram-se três formalismos com fortes semelhanças com as RdP-R: as RdP Coloridas [Jensen, 92], as RdP Interpretadas [Silva, 85] e os statecharts [Harel, 87].

Importa, nesta secção, identificar os contributos que podem ser recolhidos de cada um desses formalismos para a análise das RdP-R, no sentido da verificação de algumas das propriedades do modelo. De um modo geral, pode-se afirmar que todas as conclusões obtidas para cada um desses formalismos podem contribuir para a análise das RdP-R.

Particularizando:

- no caso dos statecharts, os trabalhos teóricos de suporte a métodos formais de verificação específicos são, praticamente, de contribuição nula;
- no caso das RdP Coloridas os resultados actualmente disponíveis são muito ricos, beneficiando do trabalho teórico previamente realizado para as RdP de baixo-nível, sendo apresentados em [Jensen, 94]. A ênfase tem sido colocada na construção do espaço de estados associado, com base no qual é possível verificar *facilmente* algumas propriedades, como limitação na marcação, vivacidade, estados de bloqueio e de retorno. Técnicas de aplicação restrita, como as baseadas em “componentes fortemente ligados” e “espaços de estados condensados” são úteis para modelos com características especiais (simetrias, por exemplo). Como principal limitação à sua utilização generalizada, refira-se o problema da explosão do espaço de estados, podendo tornar a sua análise intratável. De um modo geral, as razões da ênfase no espaço de estados devem-se a:
  - *facilidade* e *rapidez* de utilização, dado poderem ser completamente automatizadas;
  - serem completas, isto é, podem ser utilizados para verificar todo o tipo de propriedades dinâmicas;
  - difícil determinação de invariantes (embora também eles fornecedores de “todo” o tipo de propriedades).
- para o caso das RdP Interpretadas, como normal nas RdP Não-Autónomas, é possível utilizar, de forma parcial os resultados obtidos para as RdP Autónomas (ver [Silva, 85], por exemplo). De um modo geral, o espaço de estados da RdP Não-Autónoma será um

subespaço do obtido relativamente à sua RdP Autónoma associada, pelo que algumas propriedades da RdP Não-Autónoma podem ser determinadas com base no espaço de estados correspondente à RdP Autónoma, nomeadamente a limitação na marcação.

Desta forma, no que se refere à identificação dos métodos de análise aplicáveis às RdP-R, a construção do espaço de estados apresenta-se como o único método com capacidade de responder a qualquer pergunta sobre o comportamento do sistema.

Sendo um método exaustivo, tem como principal desvantagem, de forma geral, o problema da dimensão associada ao espaço de estados.

O facto de não se estar a considerar nenhuma limitação ao modelo a especificar em termos de subclasses de RdP, aliado à adição de características não-autónomas estritamente necessárias para a modelação adequada dos processos físicos que se pretendem controlar, com a correspondente alteração das semânticas de disparo das transições, impede a aplicação cabal das técnicas comuns de análise de RdP Autónomas, nomeadamente as potentes técnicas baseadas em invariantes.

Esta conclusão está completamente em sintonia com as conclusões comumente referidas na bibliografia da área, em que, de modo aproximado, as capacidades de modelação da classe e o número de propriedades determináveis utilizando as técnicas comuns de análise se consideram inversamente proporcionais.

### *3.1.3.b Estratégia para a geração de código*

No que se refere ao segundo aspecto referido, da geração do código para suporte à execução ou simulação, a atitude proposta utiliza a tradução directa da especificação ou grafo de estados associado, permitindo a geração automática de código.

Desta forma é possível estabelecer uma correspondência directa entre a especificação RdP-R (ou do grafo de estados associado) e o código a utilizar na implementação. Garante-se, assim, que todas as conclusões retiradas em tempo de análise em termos de propriedades comportamentais verificadas, se irão aplicar em tempo de execução do código gerado.

Para que o processo de geração de código para execução seja passível de uma automatização completa, torna-se necessário embeber na especificação as indicações necessárias para a resolução automática de conflitos. As técnicas propostas com este objectivo serão apresentadas detalhadamente em secção posterior deste capítulo.

A utilização de uma estratégia de geração automática do código a utilizar na execução da especificação produzida permite, claramente, garantir níveis de qualidade elevada no software produzido, determinantes da sua utilização em ambientes de desenvolvimento de sistemas e em situações concretas de engenharia.

Interessa referir que, em relação à Figura 3-3 atrás apresentada, os protótipos disponíveis no grupo de investigação em que o presente trabalho se desenvolveu, cobrem, parcialmente, os fluxos verticais indicados. Especificamente:

- em relação à fase de “especificação e verificação”:
  - o trabalho de Mestrado apresentado em [Barros, 96], permite, em relação a uma subclasse das RdP-Reactivas, as CpPNeTS, determinar o grafo de estados associado a uma especificação;
- em relação à fase de “simulação e execução” das RdP-R:
  - no que se refere ao fluxo “representação da RdP-R → código executável”:
    - a tradução de uma especificação CpPNeTS em C++ e posterior execução após compilação são disponibilizadas no trabalho apresentado em [Barros, 96];
  - no que se refere ao fluxo “máquina de estados → código executável”:
    - o trabalho de Mestrado referido em [Soares, 97], permite a execução de máquinas de estado (obtidas após a compilação para uma máquina de estados plana de uma especificação statechart).

### **3.2 Sobre ambientes de desenvolvimento**

Sem dúvida que a disponibilidade de um ambiente de desenvolvimento integrado determina, num plano académico, não apenas o nível de trabalho desenvolvido em relação a uma determinada classe de RdP, mas também se constitui em motor de desenvolvimento para novos trabalhos.

Tomando como exemplo o ambiente Design/CPN e os números disponíveis, quer de publicações, quer de trabalhos de Mestrado utilizando RdP Coloridas, fica evidente esta dependência e plenamente demonstrado o interesse, do ponto de vista académico, em dispôr de um ambiente integrado de especificação e verificação.

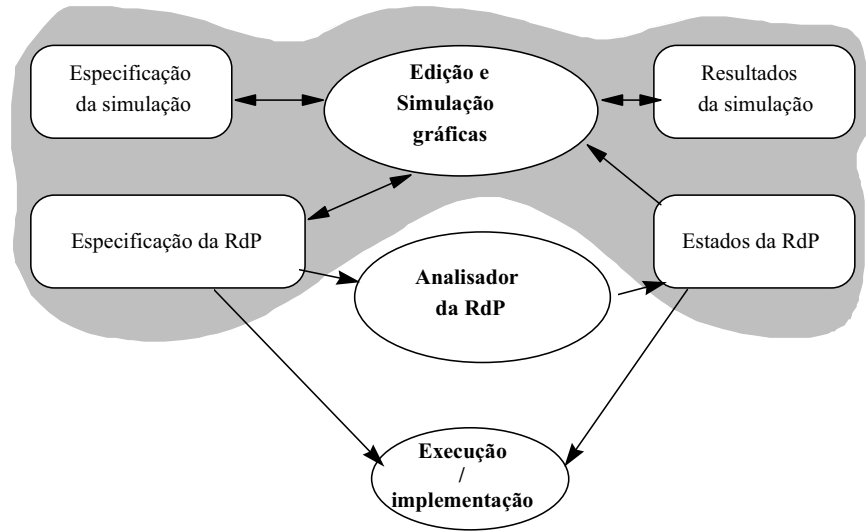


Figura 3-4 - Protótipo de ambiente de desenvolvimento [Gomes & al, 94b].

Também neste trabalho, utilizando, mais uma vez, as RdP Coloridas como uma referência para justificar as decisões tomadas, se pretendeu desde as fases menos amadurecidas do trabalho, disponibilizar um ambiente de desenvolvimento que permitisse a edição gráfica da especificação, bem como proceder à sua análise e execução. O trabalho apresentado em [Gomes & al, 94b] refere-se a um ambiente gráfico criado com esse objectivo e integra propostas e trabalhos anteriormente apresentados em [Gomes & Steiger-Garção, 92] e [Gomes & al, 93], bem como contributos de três trabalhos finais de Licenciatura em Engenharia Informática, respectivamente [Luís & Ventura, 92], [Gama & Correia, 92] e [Barros, Silva & Soares, 93].

O ambiente de desenvolvimento referido está disponível em plataformas IBM-RISC 6000 em AIX, não tendo, no entanto, conseguido passar do nível de protótipo que, globalmente, se poderá classificar de pouco robusto, isto é, utilizável com modelos de pequena dimensão (devido, fundamentalmente, ao nível de depuração de erros verificado no processo responsável pela análise da RdP). Utiliza três processos principais, como apresentado na Figura 3-4, e as comunicações entre eles são garantidas através de ficheiros de texto.

Para a representação da especificação da RdP utilizou-se o formato EDIF (*Electronic Data Interchange Format*), desenvolvido para suportar a transferência de informação sobre sistemas electrónicos entre ferramentas de automação de projecto de sistemas electrónicos. O formato contém diferentes vistas sobre o sistema a descrever; nas vistas obrigatórias (normalizadas) incluem-se a identificação dos componentes (lugares, transições e células, isto é, macronós, no caso em análise) e das interligações entre eles (arcos, no caso em análise); nas vistas adicionais, previstas no texto da norma para serem definidas para suportarem necessidades específicas de ferramentas a desenvolver (como o ambiente aqui descrito), incluem-se as descrições sobre a representação gráfica, bem como as características específicas das RdP.

O processo “Analisador da RdP” é responsável pela criação do grafo de estados associado ao modelo especificado, não sendo, no entanto, responsável pela sua interpretação, isto é, por qualquer análise de propriedades.

O processo “Edição e Simulação Gráficas” é o responsável pelo interface com o utilizador e pela coordenação de processos. Foi desenvolvido com o ambiente de geração de interfaces gráficas AIC, da IBM; a descontinuidade verificada entre versões deste produto e a cessação da sua disponibilidade provocaram o cancelamento de eventuais desenvolvimentos internos posteriores.

A Figura 3-5 apresenta um ecrã típico de edição de uma RdP, sendo visíveis algumas das facilidades de edição disponíveis. A representação gráfica das várias características do modelo RdP é realizada em caixas de texto associadas aos nós do grafo, contendo a informação específica. É suportada a representação de macrolugares e macrotransições, entendidas como páginas separadas a inserir na página hierarquicamente superior através de nós especiais, denominados por portos de entrada e portos de saída. Em capítulo posterior, o tema da representação hierárquica será detalhadamente analisado.

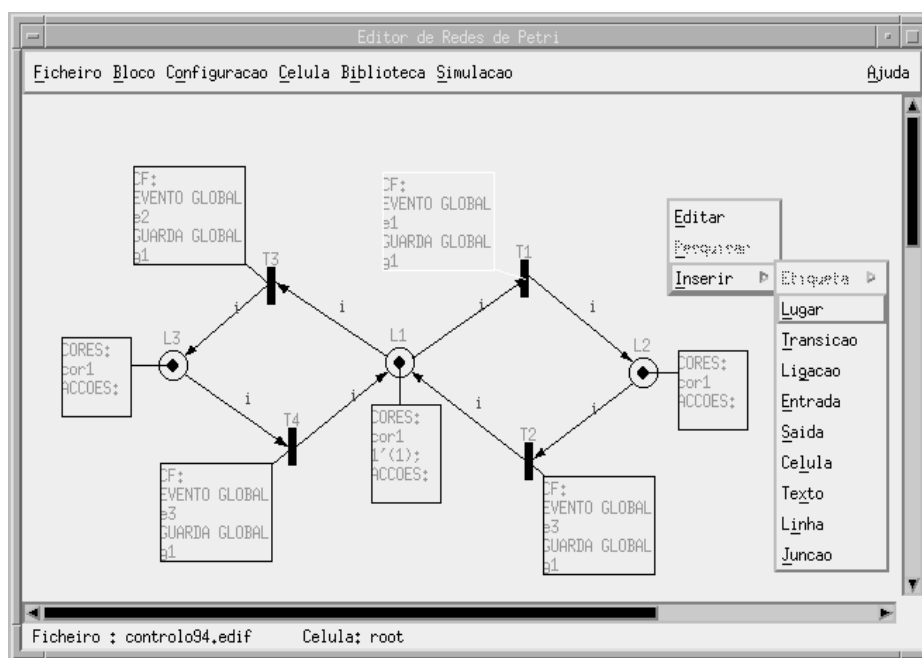


Figura 3-5 - Ecrã de edição típico (em [Gomes & al, 94b]).

No sentido de permitir a validação da especificação, uma das opções do ambiente em referência permite a simulação da execução do modelo especificado, em termos das suas entradas e saídas. Deste modo, o modelo RdP subjacente é mantido “escondido” do utilizador, sendo unicamente visíveis as relações entre as entradas e as saídas do sistema modelado (atitude de modelação tipo “caixa preta”).

É possível especificar sequências de teste, em função dos eventos de entrada, e obter os estados das variáveis de saída, em função das acções geradas. Quer a especificação da simulação, quer os resultados associados são armazenados em ficheiros de texto (ver Figura 3-4).

Foi definida uma linguagem de simulação muito simples, constituída basicamente por afectações e semelhante à utilizada pela ferramenta para programação de dispositivos de lógica programável PALASM [PALASM, 90]. A execução da especificação de simulação é realizada sequencialmente, sendo o modelo RdP executado com base num ciclo de tratamento (como referido no capítulo anterior). Na Figura 3-6(a) apresenta-se uma especificação simples, extraída de [Gomes & al, 94b], e na Figura 3-6(b) realiza-se a apresentação dos resultados da simulação resultante.

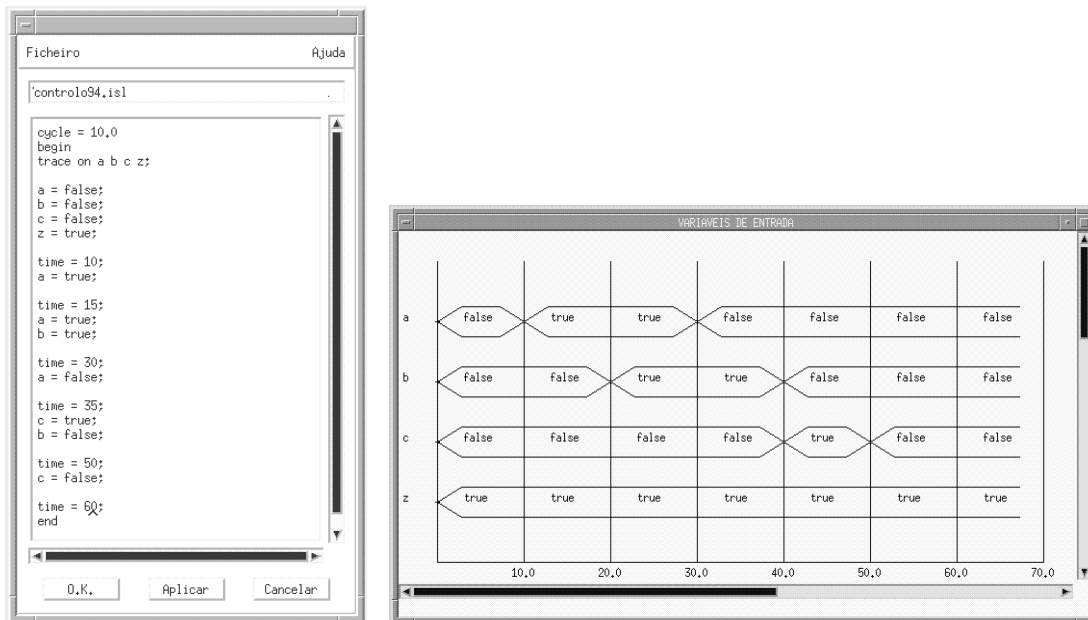


Figura 3-6 - Especificação para simulação e resultado (em [Gomes & al, 94b]).

### 3.3 Sobre a construção e utilização do grafo de estados

Como claramente identificado anteriormente, a construção do grafo de estados é de primordial importância, quer para a análise das RdP-R (sendo o único método de verificação utilizável), quer para a sua implementação em plataformas de baixo custo.

A presente secção tem como objectivo descrever aspectos relativos à sua construção e implementação em controladores industriais. No final aborda-se a necessidade de utilização de formas compactadas para a sua representação e execução, tendentes a gerir o fenómeno da explosão do espaço de estados, principal factor limitativo à utilização do grafo de estados como base para a implementação.

### 3.3.1 Das RdP-R a máquinas de estados finitos

De acordo com o descrito nas secções anteriores, o grafo de estados associado a um modelo RdP-R permitirá representar o seu comportamento, em termos dos estados de marcação da rede e das dependências nos sinais e eventos de entrada.

Utilizando um exemplo simples para ilustrar os procedimentos necessários para a construção do grafo de estados, considere-se um sistema com duas entradas binárias ( $a$  e  $b$ ) e duas saídas binárias ( $c$  e  $d$ ). As duas saídas  $c$  e  $d$  permitem memorizar a actuação das entradas  $a$  e  $b$ , respectivamente. O modelo da Figura 3-7(a) representa o comportamento descrito (excluindo a actuação das saídas).

Considerando a marcação inicial representada, os estados de marcação alcançáveis são os indicados na tabela da Figura 3-7(b); o grafo de estados associado, semelhante a uma máquina de estados como descrita em capítulo anterior, encontra-se representada na Figura 3-7(c). Para cada estado indicado devem ser consideradas todas as combinações de eventos de interesse; para cada combinação de eventos, o estado de marcação seguinte é determinado de acordo com as regras de disparo.

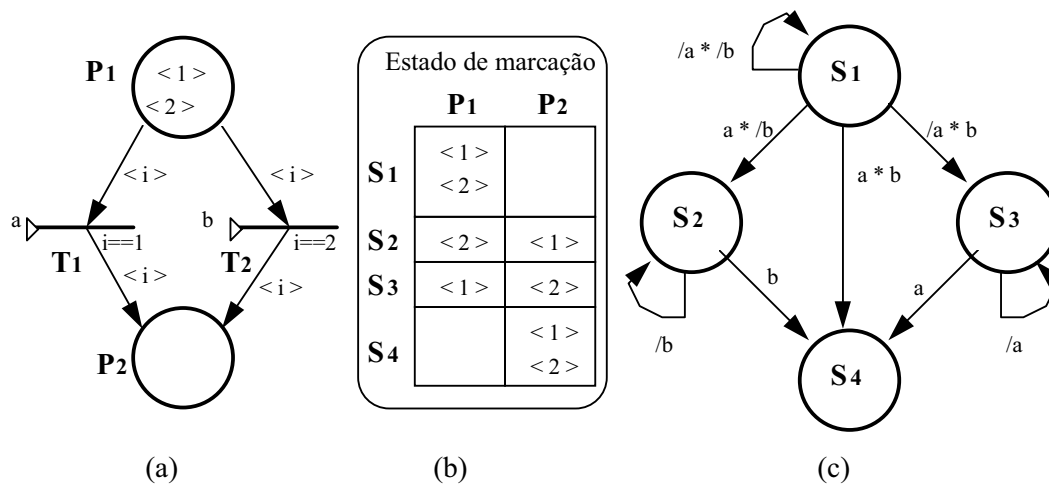


Figura 3-7 - (a) uma RdP-R simples; (b) estados de marcação possíveis; (c) grafo de estados associado.

Considerando que se representam as entradas  $a$  e  $b$  através de um vector  $e[1..2]$ , e as saídas  $c$  e  $d$  por um outro vector  $s[1..2]$ , o modelo RdP-R equivalente pode sobrepor as duas transições utilizadas na Figura 3-7(a) numa única, com arcos semelhantes aos representados, sem guarda e com um evento associado do tipo  $e[i]$ . A actuação das saídas, pode ser modelada através de uma acção associada ao lugar P2 do tipo  $IF <i> THEN s[i]$ . Obviamente que o espaço de estados associado a este segundo modelo é igual ao do primeiro modelo, assim como a máquina de estados associada. O modelo apresentado na Figura 3-7(a) pode ser obtido recorrendo aos procedimentos apresentados na subsecção 2.2.11 do capítulo anterior “Interdependência entre as partes autónoma e não-autónoma”.

Apresentado este pequeno exemplo introdutório, aborda-se um algoritmo, proposto em [Gomes & Steiger-Garção, 95d], para construção do grafo de estados. Os procedimentos podem ser agrupados em três níveis:

- considera-se que todas as transições que utilizem interdependências entre as partes autónoma e não-autónoma do modelo, são encaradas como representando um conjunto de transições em que essas dependências foram eliminadas, recorrendo aos procedimentos apresentados na já referida subsecção 2.2.11 do capítulo anterior;
- iniciar a análise pelas características não-autónomas do modelo, permitindo identificar submodelos autónomos associados a um conjunto de eventos válidos;
- determinar os estados de marcação seguintes através da análise dos modelos autónomos obtidos; as propostas apresentadas em [Huber et al., 86] constituem a principal referência.

O algoritmo proposto é apresentado seguidamente:

```

NãoProcessados = { NovoNó( m0 )}
Processados = ∅
repetir
  Seleccionar um nó x ∈ NãoProcessados
  se mx == my para algum y ∈ Processados
  então Marcar x equivalente a y
  senão
  início { x não tem nó equivalente }
    para todas as combinações de condições com eventos EC1,...,ECn, num total de 2n
    fazer
      início
        ModeloTemp = CriarModeloAutónomo ( OModelo,
          lista_de_transições_com_as_condições_associadas_válidas)
        DeterminarPróximaMarcaçãoDoModeloAutónomo ( ModeloTemp, mx,
          mv, VariáveisDeVínculo)
        v = NovoNó ( mv )
        NãoProcessados = NãoProcessados + v
        NovoArco ( x, v, lista_das_condições_associadas_válidas )
      fim
    fim
  NãoProcessados = NãoProcessados - x
  Processados = Processados + x
até NãoProcessados == ∅.

```

O algoritmo funciona do seguinte modo: cada estado de marcação é considerado um nó do grafo a gerar; considera-se uma lista de nós processados e uma lista de nós não processados; o ciclo principal inicia-se pela selecção de um nó da lista dos não processados e conclui-se com a transferência desse

nó seleccionado para a lista dos processados; determinam-se todas as combinações possíveis de eventos associados, isto é, considerando a lista de eventos associados às transições  $EC_1, \dots, EC_n$ , determinam-se as 2<sup>n</sup> combinações; cada uma delas será analisada individualmente (na implementação realizada em [Barros, 96], em relação às CPpNeTS, só se consideram os eventos associados a transições habilitadas, permitindo uma optimização em termos do tempo de processamento necessário); para cada combinação de eventos é criado um modelo autónomo temporário, obtido mantendo todas as transições que possuam eventos avaliados como válidos (ou dependentes de eventos a gerar internamente ao modelo) e removendo todas as outras transições e arcos associados; o novo estado de marcação do modelo autónomo temporário é determinado, de acordo com as regras de disparo, considerando todos os disparos possíveis dentro do passo de análise; é criado um novo nó, que será adicionado à lista de nós não processados e ligado ao nó em análise por um arco a que se associa a combinação de eventos considerada.

### 3.3.2 Execução em plataformas de baixo custo

A utilização do grafo de estados como suporte à implementação das RdP-R em plataformas de baixo custo foi identificada como a solução possível para conciliar intenções potencialmente antagónicas: utilizar plataformas com reduzidos recursos de computação e utilizar um formalismo de especificação complexo, como as RdP-R.

De entre as plataformas de baixo custo, consideram-se aquelas que utilizam linguagens booleanas específicas para a sua programação. Concretamente, consideram-se os controladores lógicos programáveis industriais, que recorrem a linguagens como os diagramas de contactos, SFC ou Grafset (ver [IEC, 92] [Grafset, 92]) e os dispositivos de lógica programável, utilizando linguagens como o PALASM (ver [PALASM, 90]). Em [Jafari & Boucher, 94] e em [Silva,85] [Tomé, 89] [Fernandes & Proença, 94] [Fernandes et al., 95] [Adamsky & Monteiro, 95] [Adamsky & Monteiro, 96] apresentam-se técnicas específicas, aplicáveis nas duas áreas referidas, permitindo a implementação de RdP de baixo nível seguras.

O grafo de estados obtido na subsecção anterior pode ser considerado como um caso particular das situações referidas, pelo que as técnicas propostas em [Jafari & Boucher, 94] e em [Silva, 85] [Tomé, 89] são aplicáveis.

Motivado por questões de completação da exposição, apresentam-se seguidamente os procedimentos que permitem a implementação do grafo de estados obtido. Como normal em sistemas digitais de pequena complexidade, utilizar-se-á um conjunto de elementos de memória, organizados num vector, para armazenar o estado actual, e lógica booleana para determinar a evolução do sistema. O vector de

elementos de memória representa os diferentes nós do grafo, numa relação biunívoca, de modo semelhante ao utilizado nos SFC/Grafcet. O elemento de memória associado ao estado activo armazenará o estado lógico “1”, enquanto todos os outros elementos conterão “0”.

Uma mudança de estado traduzir-se-á na desactivação do elemento de memória representando o estado anterior e na activação do elemento de memória associado ao estado seguinte, de acordo com a avaliação das condições de transição.

Considerando que o elemento de memória dispõe de entradas de activação e desactivação (“*set*” e “*reset*”, respectivamente), tudo se resume a construir as expressões lógicas associadas.

A forma de determinação das expressões das entradas “*set*” e “*reset*” depende do tipo dessas entradas, nomeadamente da sua característica síncrona ou assíncrona.

Caso se disponha de entradas assíncronas, no sentido de garantir um funcionamento correcto mesmo com tempos de atraso distintos para os diversos blocos, a desactivação dos elementos de memória associados ao estado anterior deverá ser realizada pela activação efectiva dos elementos de memória associados ao estado seguinte. Assim, a expressão de “*reset*” para um estado anterior é obtida pela composição OU de todos os estados seguintes, enquanto que a expressão de “*set*” para um estado seguinte se obtém pela composição OU de expressões dependentes do estado anterior e da condição que permite a transição.

Caso se disponha de entradas síncronas, a activação do estado seguinte e desactivação do estado anterior é realizada por um sinal único, no mesmo instante de análise. Denomina-se por condição de transição a expressão dependente do estado anterior e da condição que permite a transição. A expressão de “*reset*” para um estado anterior é obtida pela composição OU de todas as condições de transição envolvendo esse estado. A expressão de “*set*” para um estado seguinte é obtida pela composição OU de todas as condições de transição que conduzem à activação desse estado.

Considere-se o exemplo elementar do diagrama de estados da Figura 3-8, apresentado em [Gomes & Steiger-Garção, 95c], para ilustrar o exposto (só são referidos os eventos aos quais os estados representados são receptivos, isto é, que provocam alteração do estado).

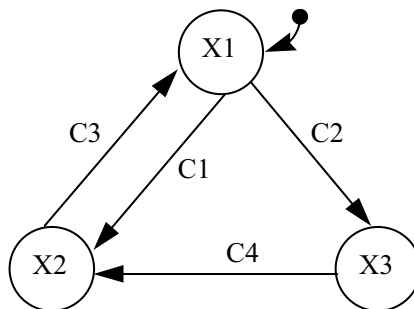


Figura 3-8 - Um diagrama de estados.

A sua análise directa permite a construção da Tabela 3-1. A coluna referida por “Expressão A” apresenta as parcelas a considerar na expressão do “reset” assíncrono de estado anterior, enquanto a coluna referida por “Expressão B” apresenta as parcelas a considerar nas expressões do “reset” síncrono de estado anterior e de “set” (síncrono ou assíncrono) de estado seguinte.

| Estado Anterior | Estado Seguinte | Condição de transição | Expressão A | Expressão B |
|-----------------|-----------------|-----------------------|-------------|-------------|
| X1              | X2              | C1                    | X2          | C1 . X1     |
| X1              | X3              | C2                    | X3          | C2 . X1     |
| X2              | X1              | C3                    | X1          | C3 . X2     |
| X3              | X2              | C4                    | X2          | C4 . X3     |

Tabela 3-1 - Análise da dinâmica do diagrama de estados da Figura 3-8.

As expressões apresentadas na Tabela 3-2 para os “set” e “reset”, síncronos e assíncronos, de cada um dos elementos de memória são determinadas directamente a partir das colunas correspondentes da Tabela 3-1 (ResetLi/SetLi e ResetFFi/SetFFi referem-se às entradas assíncronas e síncronas, respectivamente, correspondentes ao elemento de memória associado a Xi):

| Solução assíncrona        | Solução síncrona             |
|---------------------------|------------------------------|
| ResetL1 = X2 + X3         | ResetFF1 = C1 . X1 + C2 . X1 |
| ResetL2 = X1              | ResetFF2 = C3 . X2           |
| ResetL3 = X2              | ResetFF3 = C4 . X3           |
| SetL1 = C3 . X2           | SetFF1 = C3 . X2             |
| SetL2 = C1 . X1 + C4 . X3 | SetFF2 = C1 . X1 + C4 . X3   |
| SetL3 = C2 . X1           | SetFF3 = C2 . X1             |

Tabela 3-2 - Expressões de activação e desactivação.

A Figura 3-9 representa graficamente as dependências identificadas, utilizando a notação das linguagens de contactos, isto é, considerando uma implementação baseada em controladores programáveis industriais, admitindo-se a disponibilidade de elementos de memória com entradas assíncronas de activação e desactivação. Considera-se que as entradas “set” têm prioridade sobre as entradas “reset”.

No exemplo apresentado, foi omitida a parte responsável pela inicialização da máquina de estados.

Na Figura 3-10 apresenta-se uma especificação em PALASM, utilizando o dispositivo de lógica programável PALCE16V8, de utilização muito vulgar. Recorre à especificação explícita das equações booleanas indicadas, de modo semelhante ao realizado na Figura 3-9, considerando, neste caso, entradas síncronas e que a saída de um elemento de memória pode ser especificada através da expressão

$$Q_{\text{seguinte}} = (Q_{\text{anterior}} + \text{Set}) * /\text{Reset}.$$

A inicialização é garantida através da entrada “init”. Neste caso, as capacidades intrínsecas da linguagem poderiam ter sido utilizadas permitindo a especificação directa da máquina de estados.

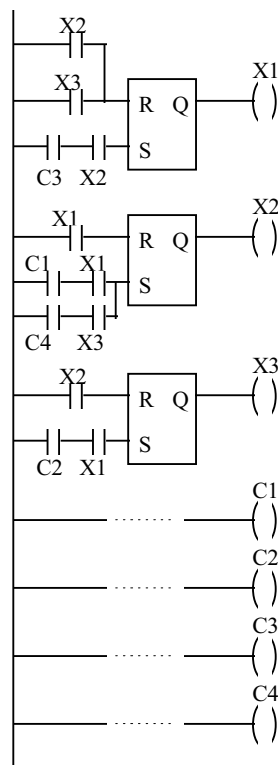


Figura 3-9 - Diagrama de contactos associado ao grafo de estados da Figura 3-8.

```

...
CHIP _exemplo1 PALCE16V8
;----- PIN Declarations -----
PIN 1      clock      ;
PIN 2      init       ;
PIN 3      c1         ;
PIN 4      c2         ;
PIN 5      c3         ;
PIN 6      c4         ;
PIN 11     oe         ;
PIN 14     x1         REGISTERED ;
PIN 15     x2         REGISTERED ;
PIN 16     x3         REGISTERED ;
;----- Boolean Equation Segment -----
EQUATIONS
x1 := init + (x1 + c3*x2) * /((c2 + c1)*x1) ;
x2 := (x2 + c1*x1 + c4*x3) * /(c3*x2 + init) ;
x3 := (x3 + c2*x1) * /(c4*x3 + init) ;
...

```

Figura 3-10 - Especificação Palasm associada à máquina de estados da Figura 3-8.

### 3.3.3 Obtenção de grafos de estados condensados

Como referido no início da presente secção, o grafo de estados associado a uma RdP é um conceito importante, sob vários pontos de vista, nomeadamente para a sua análise e implementação. Enquanto o primeiro aspecto (análise) é valorizado em extensa bibliografia e ferramentas de análise, permitindo a verificação da especificação, o segundo aspecto (implementação) foi objecto de particular atenção neste trabalho.

De uma forma geral, as técnicas de exploração do espaço de estados têm vindo a ser utilizadas como ferramenta de depuração e verificação de correcção da especificação de sistemas concorrentes [Godefroid, 96]. A razão deste “sucesso” pode atribuir-se à simplicidade destas técnicas; são fáceis de entender, implementar e utilizar, dado poderem ser completamente automatizadas.

Como também referido anteriormente, uma das principais desvantagens deste método reside no número potencialmente muito elevado de estados resultante da análise de um modelo RdP *medianamente* complexo.

Torna-se, desta forma, necessário utilizar formas compactadas para a sua representação, tendentes a gerir o fenómeno da explosão do espaço de estados. Também estas formas compactadas de representação podem ser vocacionados para o suporte à análise da especificação ou para a sua execução.

Os trabalhos conhecidos são vocacionados para o suporte às tarefas de análise da especificação, ou seja, de verificação de propriedades. De entre eles, refiram-se os (que se consideram) mais significativos:

- no âmbito dos trabalhos em RdP Coloridas, de particular relevância, refira-se o livro de Kurt Jensen [Jensen, 95] em que se aborda a problemática da análise das RdP Coloridas através de grafos de ocorrências, grafos de ocorrências com classes equivalentes e de grafos de ocorrências com simetrias; a eficácia das técnicas propostas, aplicáveis apenas a RdP Autónomas, em termos do factor de redução do número de estados do grafo, está fortemente relacionado com o tipo de modelo em análise;
- no âmbito dos métodos formais de verificação de sistemas, refiram-se os trabalhos de A. Valmari [Valmari, 91], onde se propõem os “*stubborn sets*”, e de Patrice Godefroid [Godefroid, 96], onde se introduzem os “*sleep sets*”.

Não se conhecem trabalhos específicos abordando a problemática da compactação do espaço de estados vocacionada para o suporte à execução. É, claramente, um tema necessitando de trabalho de investigação específico; constitui uma das lacunas identificadas no decurso deste trabalho, merecedora, na opinião do autor, de trabalhos de investigação posteriores. Para o caso particular das RdP-R, considerando o papel que a execução do grafo de estados pode assumir, a disponibilização de técnicas específicas para a sua compactação é particularmente interessante.

Um caso específico, referido apenas como exemplo, em que se torna particularmente relevante a compactação do grafo de estados, ocorre quando é possível decompor o estado de marcação da RdP-R em várias componentes autónomas ou fracamente ligadas, isto é, que evoluem independentemente ou com pequenas situações de dependência mútua; deste modo, será possível evitar a necessidade de produção de um espaço de estados resultado do produto cartesiano dos espaços de estados parcelares (produzindo-se, em sua substituição, um espaço de estados associado à sincronização entre os espaços de estados parcelares). Será esse o caso que se encontrará quando, em capítulo posterior, forem propostas técnicas de modelação de controladores difusos através de RdP-R.

### **3.4 Resolução de conflitos**

A resolução de conflitos apresenta-se como um importante tema a ser tratado no sentido do suporte à produção automática de código directamente a partir do modelo RdP-R. Em diversas áreas de aplicação, a atitude do projectista considerada adequada é a de evitar conflitos (na medida em que os conflitos permitem modelar não-determinismos na execução do modelo). Veja-se, por exemplo, [Silva, 85] em que essa atitude é defendida. Contudo, os conflitos estão presentes nos sistemas que se

pretendem modelar e a sua inclusão na RdP-R pode tornar-se uma necessidade absoluta. Casos típicos incluem semáforos de exclusão mútua e partilha de recursos comuns.

Para evitar a intervenção do “utilizador”, torna-se necessário definir *a priori* um conjunto de critérios que permitam resolver de forma automática os conflitos presentes na especificação.

Independentemente da execução do modelo ser realizada através de um jogador-de-marcas ou de um jogador-de-estados, a definição desse conjunto de critérios para a resolução automática de conflitos pode conduzir a uma execução determinística e “viciada” do modelo, em que a resolução dos conflitos não se realiza de forma equitativa.

Tome-se um exemplo simples de duas transições em conflito T1 e T2, ligadas a um mesmo lugar P1. Um critério determinístico a embeber no modelo de RdP-R a utilizar (por exemplo a utilização de prioridades entre as transições), conduzirá, normalmente, à não verificação de algumas propriedades importantes (como a de RdP viva, dado que, no caso do exemplo referido, uma das transições nunca disparará quando se verificar o conflito).

Nesta secção será apresentada uma taxonomia para a resolução de conflitos, bem como os conjuntos de procedimentos associados e recursos necessários para a sua execução, inicialmente proposta em [Gomes & Steiger-Garção, 96b].

O objectivo desta sistematização é o de permitir ao projectista a selecção de um determinado critério, suportando a modelação de não-determinismo e tendo presente os recursos necessários e condicionantes associados.

Os procedimentos propostos incluem o recurso a modelos com prioridades associadas às transições e o uso de sub-redes (embebidas no modelo inicial ou externas a este) ou de sistemas externos responsáveis pela resolução dos conflitos. Estas sub-redes actuam como árbitros especializados na resolução de um determinado conflito.

### **3.4.1 Classificação das soluções propostas**

Diversas são as abordagens possíveis ao problema da classificação das soluções propostas para a resolução automática de conflitos. No que se segue utilizam-se, como referências de enquadramento, três dicotomias largamente conhecidas e que permitem dividir o espaço de soluções.

As dicotomias utilizadas são as de “sistemas determinísticos versus não-determinísticos”, de “resolução de conflito equitativa versus não-equitativa” e de “execução baseada num jogador-de-marcas versus jogador-de-estados”.

Inicia-se a análise considerando a dicotomia “sistemas determinísticos versus não-determinísticos”. Nos sistemas determinísticos, o próximo estado/acção é definido univocamente; corresponde à atitude comum nos formalismos normalmente utilizados na modelação de controlo de sistemas a eventos discretos. No entanto, como atrás referido, é, frequentemente, conveniente a modelação de não-determinismo (em termos da terminologia utilizada na comunidade de RdP, escolhas, conflitos e confusões). Exemplos do dia a dia incluem a modelação/simulação de comunicações em ambientes de manufactura, em que as redes disponíveis impõem um atraso variável na transmissão de mensagens, bem como a modelação de estratégias de suporte à partilha de recursos (exclusão mútua ou partilha rotativa, por exemplo).

A segunda dicotomia, “resolução de conflito equitativa versus não-equitativa”, é de importância vital para as características comportamentais a exibir pelo modelo. Num conflito equitativo, nenhuma transição (ou grupo de transições) pode(m), de modo permanente, negar a possibilidade de qualquer outra transição envolvida disparar. O índice de equidade associado a uma determinada transição é, aqui, definido como a relação entre o número de disparos efectivos e o número de situações de disparo potencial (isto é, de habilitações) verificado nessa transição.

A terceira dicotomia caracteriza a forma como o modelo de RdP-R é executado, nomeadamente através de um jogador-de-marcas ou da execução do grafo de estados associado. Como referido em secção anterior, as implementações baseadas na execução do espaço de estados são de especial interesse quando a plataforma de implementação possui recursos computacionais *muito* limitados, como é o caso de controladores industriais de baixo custo e de alguns dispositivos de lógica programável.

Em termos gerais, as soluções a apresentar para a resolução automática de conflitos podem ser classificadas em um de dois tipos:

- no primeiro, realizam-se modificações no modelo inicial, de modo a minimizar as alterações das suas características comportamentais,
- no segundo recorre-se à utilização de modelos adicionais com execução paralela, sincronizáveis com o modelo inicial e com capacidades de resolver o conflito.

No primeiro tipo, as técnicas de execução apresentadas em secção anterior podem ser utilizadas, nomeadamente as baseadas em jogadores-de-marcas ou em jogadores-de-estados, enquanto que no segundo tipo se elege, preferencialmente, a execução do espaço de estados.

Uma reflexão sobre esta atitude permite concluir que as resoluções de conflito que evidenciem a componente não-autónoma do modelo serão suportadas pela execução concorrente de árbitros. Caso a

ênfase seja dada à componente autónoma do modelo, nomeadamente através de modificações da rede, o método preferencial será baseado num jogador-de-marcas (embora as implementações com base na execução do espaço de estados também sejam possíveis).

Deste modo, a classificação dos tipos de resolução de conflito pode ser sintetizado através da Figura 3-11, em que se recorre a um referencial em termos de equidade versus determinismo versus execução. A exposição que se segue visa uma análise separada dos oito possíveis casos, nomeadamente as regiões I a VIII da Figura 3-11. Normalmente, os casos baseados na execução do espaço de estados serão apresentados após os correspondentes casos baseados em jogadores-de-marcas o terem sido. O significado do texto referido na Figura 3-11 e associado a cada uma das regiões será clarificado ao longo do texto que se segue.

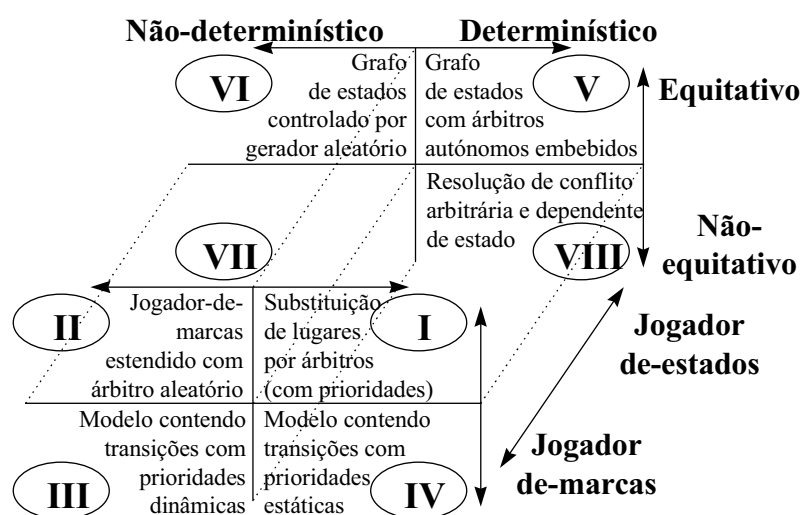


Figura 3-11 - Representação de uma taxonomia de soluções para a resolução automática de conflitos.

Os exemplos a utilizar recorrem, normalmente, a RdP-R em que apenas as características de baixo-nível são utilizadas. Contudo, os resultados são aplicáveis em RdP-R utilizando características de alto-nível, considerando que a marcação referida deve ser vista como um vínculo específico de uma marcação “colorida”.

### 3.4.1.a Resolução não-equitativa

#### 3.4.1.a.1 Utilização de prioridades associadas às transições

A análise das várias regiões da Figura 3-11 é iniciada pela que, provavelmente, será a solução mais comumente utilizada, isto é, a região determinística e não-equitativa, assinalada com IV na referida

figura. Neste caso, a solução passa pela utilização de um esquema de prioridades associadas às transições em conflito.

Embora extensivamente utilizado em ambientes de desenvolvimento disponíveis, a implementação do esquema de prioridades apresenta várias opções. A utilização de apenas dois níveis de prioridade apresenta-se como muito interessante do ponto de vista da implementação, nomeadamente da eficiência de execução do modelo. Contudo, no caso geral (com mais que duas transições em conflito) não é suficiente, pelo que um esquema com múltiplos níveis de prioridade deve ser considerado.

As estratégias consideradas para a indicação dos níveis de prioridade podem ser caracterizadas do seguinte modo:

- baseadas no tipo de ordenação praticado, podendo a ordenação ser global ou local, isto é, a associação de uma ordem de prioridades às transições pode ser realizada de modo global (considerando todas as transições no conjunto a ordenar) ou local (em que se consideram apenas as transições contidas na estrutura em conflito). Embora a segunda hipótese seja mais natural e elegante, a primeira hipótese permite a atribuição de prioridades de forma cega e automática através, por exemplo, de um mecanismo de baralhar aleatoriamente uma série de números inteiros sem repetições e posterior associação a todas as transições do modelo (o caso mais simples é a série de 1 a N, com N igual ao número de transições);
- associadas à entidade que define as prioridades. As entidades potencialmente responsáveis pela definição das prioridades são: o projectista e aplicações responsáveis pela sua geração automática (aleatoriamente ou através de outro critério definido *a priori*). É de interesse considerar a utilização de sistemas dedicados de apoio à decisão (por exemplo, um sistema pericial) com capacidades de produzir sugestões baseadas em critérios definidos a priori ou em bases de conhecimento. A sugestão produzida pode ser adoptada directamente ou validada pelo projectista ou utilizador.

Estratégias mais complexas incluem a definição de prioridades dinâmicas, podendo ser realizadas de modo determinístico ou não (associado à região III da Figura 3-11). Novamente, estas estratégias podem ser acompanhadas pela utilização de procedimentos com algoritmos embebidos (produzindo uma série de valores, por exemplo) ou dispondo de capacidades de acesso a informação (nomeadamente o estado da RdP-R em análise), recursos específicos (um árbitro externo, por exemplo) ou um sistema pericial.

Uma solução específica possível pertencente a esta região é baseada na definição de vários níveis de prioridade, em número inferior ao de transições em conflito; devido à ambiguidade resultante do

conjunto de transições em conflito com o mesmo valor de prioridade associado, a resolução pode ser resolvida através de uma escolha aleatória, sempre que o conflito se verifique.

#### **3.4.1.a.2 Construindo um espaço de estados não-equitativo**

Do ponto de vista do grafo de estados, um conflito pode ser modelado através de um estado de marcação (nó do grafo) com diversos estados seguintes possíveis (como o exemplo da RdP de escolha-livre a ser apresentado na Figura 3-12(a)).

No caso da região VIII da Figura 3-11, a resolução de conflitos determinística e não-equitativa para implementações baseadas no espaço de estados é conseguida através da construção inicial do grafo de estados e, para cada caso de conflito, seleccionar um dos próximos estados possíveis (e ignorar os restantes estados de marcação alternativos). Neste contexto, o conceito de estado está, claramente, associado a uma marcação específica da rede. Novamente, a escolha pode ser realizada aleatoriamente ou sugerida por um sistema de apoio à decisão.

De um modo geral, para um determinado estado de marcação, dependendo da combinação dos eventos externos avaliados positivamente, o conjunto dos conflitos presentes nesse estado será arbitrariamente resolvido e o novo estado será encontrado. Isto é, a resolução de um conflito é função do estado global da rede. Como consequência, o sistema comporta-se deterministicamente, isto é, para cada estado e combinação de eventos externos é identificado um estado seguinte único.

Assim, a resolução de um determinado conflito pode ser realizada de modo não uniforme quando considerado em diferentes estados globais.

A análise da região VII da Figura 3-11 mostra-se de reduzido interesse, do ponto de vista de aplicação prática, pelo que não será analisada. Corresponde a implementações não-determinísticas e não-equitativas baseadas na execução do espaço de estados. De forma aproximada, resulta da junção das características das regiões III e VIII da Figura 3-11.

#### **3.4.1.b Resolução determinística e equitativa**

A análise da Figura 3-11 prossegue com as regiões I e V, em que se adoptam estratégias determinísticas e equitativas. Nestes casos, no sentido de tornar o modelo não-determinístico inicial num outro que seja determinístico, é proposta a utilização de sub-redes, isto é, de árbitros responsáveis pela resolução do conflito. O caso geral, em que é pretendida uma equidade absoluta entre as transições envolvidas, é um problema complexo.

Assim, a análise será realizada considerando três casos:

- os dois primeiros utilizam alterações ao modelo inicial, que podem ser tornadas automáticas, de modo que um jogador-de-marcas poderá ser utilizado para executar o modelo (região I da Figura 3-11);
- o caso restante aborda as implementações baseadas na execução do espaço de estados (região V da Figura 3-11).

#### **3.4.1.b.1 O caso das RdP de escolha-livre**

O primeiro caso a abordar é o das RdP de escolha-livre. Nesta classe de redes, os vários conflitos possíveis têm complexidades similares, sendo o mais simples de apresentar correspondente a um lugar com várias transições de saída, ver Figura 3-12(a). Conflitos com mais lugares de entrada envolvidos têm os mesmos procedimentos de resolução.

Equidade e determinismo podem ser alcançados simultaneamente através da utilização de um árbitro (sub-rede adicional a embeber no modelo inicial), comportamentalmente semelhante a uma máquina de estados e em que a activação de cada estado corresponde à habilitação de uma transição específica.

O número de estados (lugares) a adicionar é, no mínimo, igual ao número de transições em conflito, ver Figura 3-12(b).

É trivial a implementação de diferentes esquemas equitativos. Por exemplo, diferentes índices de equidade associados às transições, como seja equidade ponderada com diferentes sequências de disparo possíveis (por exemplo, sequências pseudo-aleatórias com um comprimento pré-definido). A Figura 3-12(c) ilustra um exemplo em que a transição T3 foi duplicada, permitindo que o caminho associado com T3 seja escolhido o dobro das vezes que os caminhos associados com T1 e T2. Na Figura 3-12(b), os índices de equidade associados às transições são de  $1/3$ , enquanto que na Figura 3-12(c) os índices de equidade dos caminhos associados a T1, T2 e T3 são de  $1/4$ ,  $1/4$  e  $1/2$ , respectivamente. Igual resultado seria obtido recorrendo a arcos ponderados, como ilustrado na Figura 3-12(d).

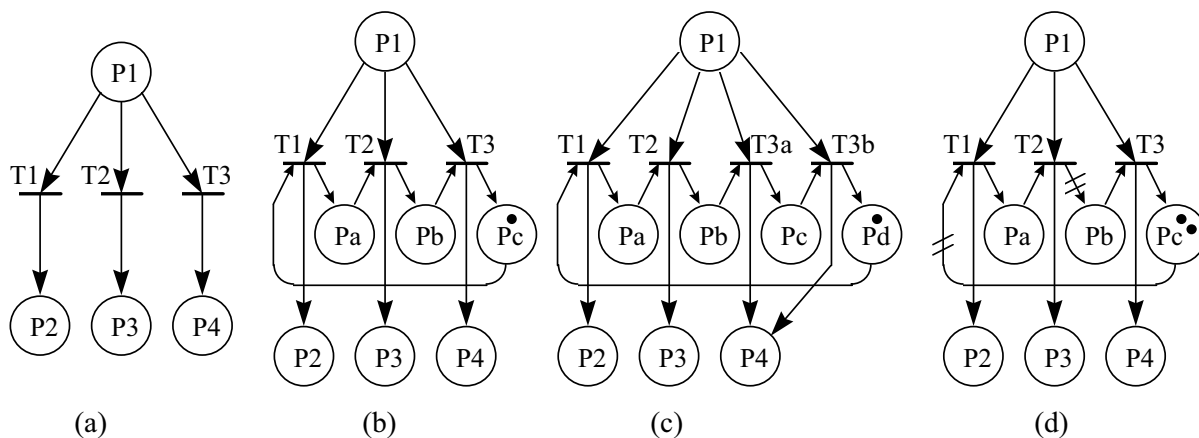


Figura 3-12 - Resolução de conflitos em RdP de escolha-livre: a) modelo inicial; b) modelo comportamentalmente equivalente sem conflitos; c) e d) introdução de diferentes índices de equidade.

Considerando o caso de utilização de características coloridas, por razões que ficarão claras no parágrafo seguinte, para que a solução proposta baseada na adição de uma máquina de estados seja utilizável, torna-se necessário que todas as marcas presentes em P1 habilitem todas as transições em conflito, ou seja, que a situação de conflito identificada se mantenha inalterada (para um determinado conjunto de transições). Neste caso, todas as inscrições associadas aos arcos e guardas associadas às transições no modelo inicial se mantêm; os novos lugares e arcos a eles interligados podem utilizar marcas elementares (não-coloridas).

Contudo, do ponto de vista do modelo não-autônomo este esquema encontra vários óbices à sua aplicação. Considerando o caso geral em que diferentes eventos se encontram associados às várias transições, diferentes situações de conflito são encontradas dependendo do conjunto de eventos avaliados positivamente; cada situação possui um diferente conjunto de transições em conflito e a solução apresentada baseada na adição de uma máquina de estados não é utilizável. No entanto, o não-determinismo representado na Figura 3-12(a) pode ser de interesse em várias situações de modelação de comportamentos, onde alternativas de controlo devem ser especificadas. As duas secções seguintes endereçam o problema geral da resolução de conflitos determinística e equitativa.

### 3.4.1.b.2 Adicionando árbitros determinísticos e equitativos

A análise inicia-se com o problema geral do semáforo de exclusão mútua, ver Figura 3-13(a).

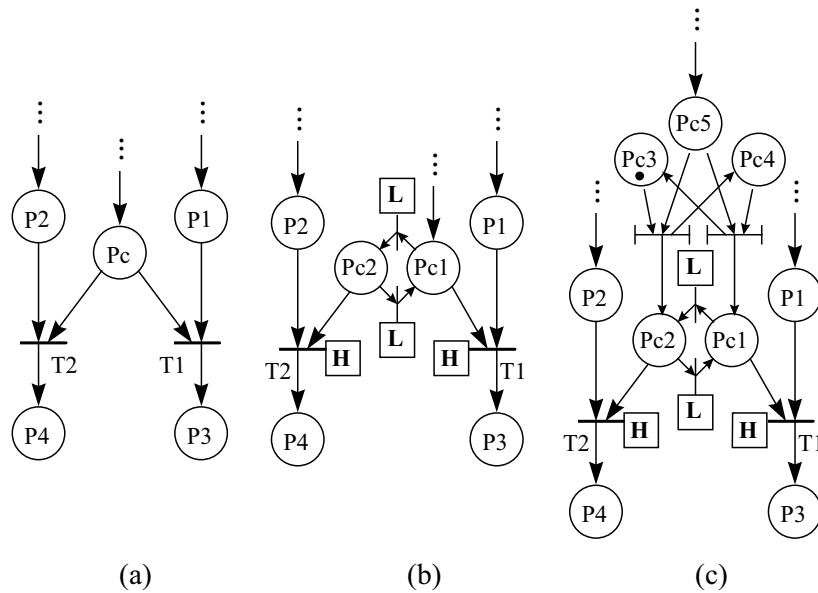


Figura 3-13 - Resolução de um conflito em problema de exclusão mútua: a) modelo inicial; b) modelo quasi-equivalente sem conflito; c) modelo quasi-equivalente sem conflito e inicialização equitativa.

Em [Smith, 95] é apresentada uma solução para o problema da exclusão mútua, ilustrada na Figura 3-13(b), contendo, em termos comportamentais, um ligeiro desvio em relação ao modelo inicial. Baseia-se na substituição do lugar comum às transições em conflito, que modela a exclusão mútua, por um árbitro (sub-rede), modelando uma máquina de estados e utilizando um esquema de transições prioritárias, com dois níveis de prioridades (designadas por *alta* e *baixa* e referidas por *H* e *L*, respectivamente). As transições internas à máquina de estados possuem prioridade baixa, enquanto que as transições em conflito possuem prioridade alta.

O esquema não é garantidamente equitativo, dado que, se P1 e P2 estiverem marcados permanentemente, sempre que uma marca “retorna” a Pc (isto é, a Pc1 na Figura 3-13(b)), o disparo de T2 nunca ocorre, sendo o modelo produzido equivalente ao modelo inicial em que se associam prioridades *alta* a T1 e *baixa* a T2, como numa solução não-equitativa.

Para evitar este comportamento, em [Gomes & Steiger-Garção, 96b] propõe-se a adição de uma segunda máquina de estados, responsável pela inicialização equitativa da máquina de estados inicial, como apresentado na Figura 3-13(c). Continuando a admitir a marcação “permanente” de P1 e P2 (sempre que Pc se encontra marcado), os índices de equidade associados aos caminhos P1-P3 e P2-P4 são iguais a  $\frac{1}{2}$ .

De notar que, se uma notação hierárquica estiver disponível, o árbitro, composto por cinco lugares (dos quais quatro novos) e quatro novas transições, pode ser encapsulado num macrolugar, permitindo uma representação próxima da utilizada na Figura 3-13(a), ao nível de topo, como apresentado na Figura 3-14.

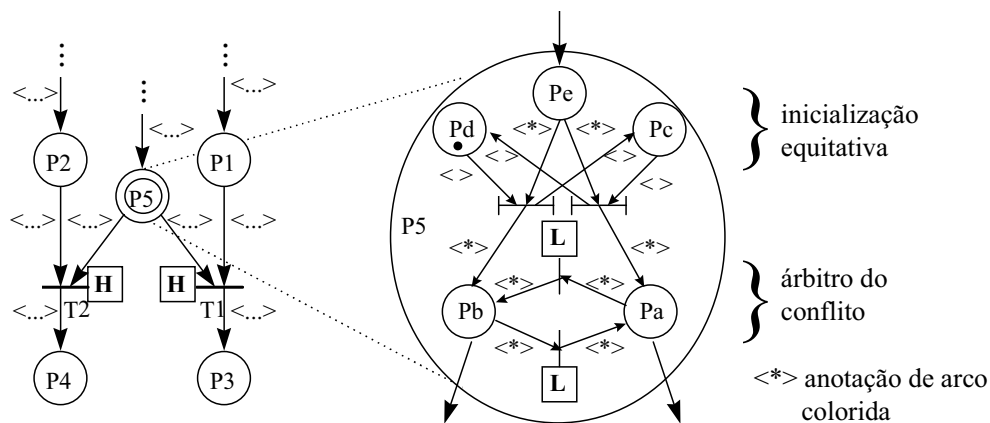


Figura 3-14 - Resolução determinística e equitativa do problema de exclusão mútua: representação hierárquica.

A extensão para redes utilizando características coloridas é trivial, como apresentado na Figura 3-14, resumindo-se aos seguintes passos (excluindo a atribuição de prioridade alta às transições em conflito):

- todas as inscrições dos arcos e guardas nas transições do modelo inicial são mantidas no representação hierarquicamente de topo (isto é, da Figura 3-13(a) para a Figura 3-14);
- os lugares da máquina de estados para inicialização equitativa (Pc e Pd na Figura 3-14) contêm marcas elementares (não-coloridas), bem como os arcos associados;
- os lugares da máquina de estados de arbitragem e de recepção das marcas externas (Pa, Pb e Pe na Figura 3-14) contêm marcas com a mesma cor que o lugar inicial (P5 na Figura 3-14); os arcos associados permitem a sua “circulação”.

Esta técnica de substituição de lugares pode ser utilizada noutras situações mais complexas, sendo, no entanto, necessário “refinar” os procedimentos no sentido de os tornar gerais.

A Figura 3-15(a) apresenta um exemplo simples de uma confusão. Na Figura 3-15(b) apresenta-se a rede resultante da aplicação dos procedimentos de substituição apresentados.

No entanto, caso a “marcação” dos macrolugares P1 e P2 conduza a uma marcação simultânea de P1b e P2b, não será possível o disparo da transição T2, mesmo considerando que os eventos associados a T1 e T3 não ocorrem. A razão deve-se ao facto de se considerar que o disparo das transições é sincronizado através de um ciclo de tratamento (relógio global que determina os instantes em que a evolução do estado do modelo se realizará), o que determina o disparo sincronizado de transições habilitadas concorrentes; no caso do exemplo da Figura 3-15(b), a marcação seguinte a P1a-P2a será

P1b-P2b e vice-versa (considerando que os eventos associados a T1 e T3 não são avaliados verdade), pelo que a transição T2 nunca disparará.

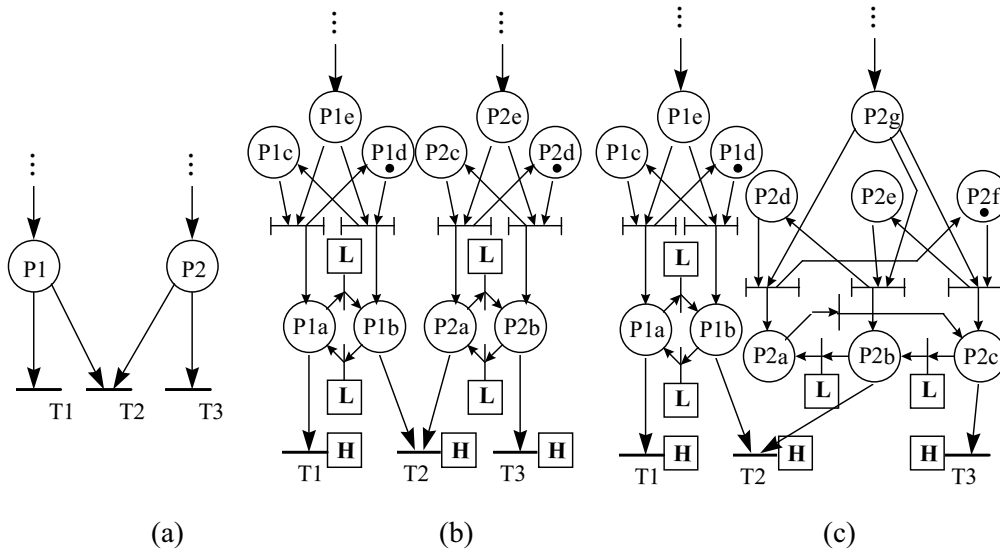


Figura 3-15 - Outra resolução de conflito: a) modelo inicial;

b) modelo não-equivalente sem conflitos; c) modelo quasi-equivalente sem conflitos.

Desta forma, introduz-se uma mudança de comportamento significativa relativamente ao modelo inicial, não sendo admissível. A solução para o problema é obtida através da utilização de árbitros com diferente número de estados, em que as máquinas de estado utilizadas para substituir os vários lugares ligados a cada uma das transições envolvidas no conflito possuam um diferente número de estados no ciclo. A Figura 3-15(c) é obtida desta forma. Este procedimento conduz à utilização de lugares “neutros”, como o lugar P2a na Figura 3-15(c). Neste exemplo, considerando que novas marcas chegam a P1e e P2g todos os ciclos de tratamento, os índices de equidade associados às transições T1, T2 e T3 são de  $5/6$ ,  $1/6$  e  $5/6$ , respectivamente. Estes índices de equidades mantêm-se, independentemente da marcação inicial da máquina de estados de inicialização do árbitro (neste caso, seis possibilidades foram consideradas).

Considerando a dependência da componente não-autónoma da rede, é importante assegurar um comportamento equitativo dos árbitros, independentemente do conjunto de eventos externos associados às transições (condição não satisfeita pela RdP da Figura 3-15(b)). Neste caso, se o evento externo associado com T1 permanecer falso, os índices de equidade associados com T2 e T3 são  $1/6$  e  $5/6$ , respectivamente (para todas as possíveis inicializações dos árbitros). Por outro lado, se o evento externo associado com T3 permanecer falso, os índices de equidade associados com T1 e T2 são  $5/6$  e  $1/6$ , respectivamente (para todas as possíveis inicializações dos árbitros). Finalmente, para o caso em que o evento externo associado com T2 permanece falso, o conflito não existe.

Seguidamente apresentam-se os passos necessários à obtenção de árbitros, a usar como sub-redes de substituição (macrolugares) dos lugares ligados a transições em conflito:

- substituir todos os lugares de entrada de transições envolvidas no conflito por um macrolugar;
- cada macrolugar é composto por três sub-redes (ou níveis) interligadas em cascata;
- o primeiro nível é composto por um lugar, funcionando como um lugar de entrada para todos os arcos do modelo inicial;
- o terceiro nível é o árbitro do conflito propriamente dito e é composto por um anel de lugares, funcionando como lugares para ligação dos arcos de saída do lugar comum no modelo inicial;
- as transições contidas no anel de lugares possuem prioridade baixa associada, enquanto que as transições inicialmente em conflito possuem prioridade alta;
- o número de lugares no anel é, pelo menos, igual ao número de transições de saída do lugar inicial; devido à adopção de sincronização global do disparo de transições por um relógio, como referido, o número deve ser aumentado de modo a satisfazer a condição de que o comprimento do anel de um árbitro ligado a uma transição não é múltiplo relativamente a outros árbitros ligados à mesma transição;
- o segundo nível garante uma inicialização equitativa do anel de arbitragem; como ilustrado, as marcas recebidas são sequencial e uniformemente distribuídas para todos os lugares no anel;
- a rede obtida deve apresentar um comportamento equitativo relativamente às transições envolvidas, em qualquer situação; esta propriedade é verificada para todas as marcações possíveis dos níveis de inicialização dos árbitros e para todas as combinações de avaliação dos eventos externos associados.

A solução proposta na Figura 3-15(c) satisfaz todos os requisitos indicados.

Estes procedimentos podem ser utilizados no sentido de produzir modelos de referência (“*templates*”) utilizáveis na substituição de lugares e na obtenção automática de modelos isentos de conflitos. Estes procedimentos podem ser facilmente embebidos no editor gráfico das RdP-R, identificando o conjunto de transições em conflito e, automaticamente ou sobre controlo do projectista, proceder à

substituição no modelo inicial de lugares por macrolugares-referência e das transições por transições com prioridade alta associada.

Trabalho posterior nesta área é plenamente justificado, nomeadamente no sentido de caracterizar estruturas alternativas utilizáveis nos árbitros, de determinar o número de lugares no anel de inicialização e no anel de arbitragem, permitindo obter índices de equidade pré-especificados (de modo semelhante ao que é possível nas RdP de escolha-livre, como apresentado). A análise de invariantes aparece como necessária à validação dos árbitros-padrão a utilizar.

### 3.4.1.b.3 Implementações baseadas no espaço de estados

Do ponto de vista das implementações suportadas por controladores de muito baixo custo, a região V da Figura 3-11 apresenta-se como a de maior interesse. Aí se endereçam as implementações baseadas na execução do grafo de estados, de forma determinística e equitativa.

Neste caso, a resolução do conflito é conseguida através da execução de um árbitro externo, implementado por uma máquina de estados executada paralelamente com a RdP-R em análise. A dinâmica desta máquina de estados é condicionada pela execução do espaço de estados associado ao modelo em análise; o seu estado será embebido no espaço de estados inicial de modo a resolver o conflito.

A junção do espaço de estados associado ao árbitro com o espaço de estados associado ao modelo inicial permitirá resolver os conflitos. No grafo de estados inicial, cada conflito é traduzido por vários arcos de saída não-determinística a partir de cada nó contendo conflitos. A estes arcos será associada uma condição dependente do estado do árbitro, eliminando o não-determinismo.

Cada conflito efectivo possuirá um árbitro associado; este árbitro é composto por, pelo menos, tantos estados quantas as possibilidades de resolução dos conflitos. Como referido no exemplo das RdP de escolha-livre, ver Figura 3-12(c), esquemas de “equidade ponderada” ou “probabilidades de disparo associadas a transições” são de implementação trivial. Por conflito efectivo entende-se uma situação específica de conflito associada a uma determinada combinação de sinais/eventos externos avaliados como existentes.

Como exemplo considere-se a Figura 3-12(a), em que se associam às transições T1, T2 e T3 os eventos externos e1, e2 e e3, respectivamente. Partindo de um estado representando a marcação isolada de P1, devem ser considerados oito arcos de saída. O primeiro corresponde a uma avaliação como falsos de todos os eventos associados; não se disparará nenhuma transição e o estado mantém-se. Outros três arcos de saída estão associados a situações em que apenas um dos eventos é avaliado

como verdadeiro e em que não ocorrem conflitos. Os quatro arcos restantes correspondem a quatro situações de conflito efectivo, que necessitam de ser resolvidos por árbitros dedicados.

O processo é ilustrado na Figura 3-16, utilizando novamente o exemplo do semáforo de exclusão mútua. As saídas geradas pela execução do modelo são representadas após uma barra (como em XXX/saída).

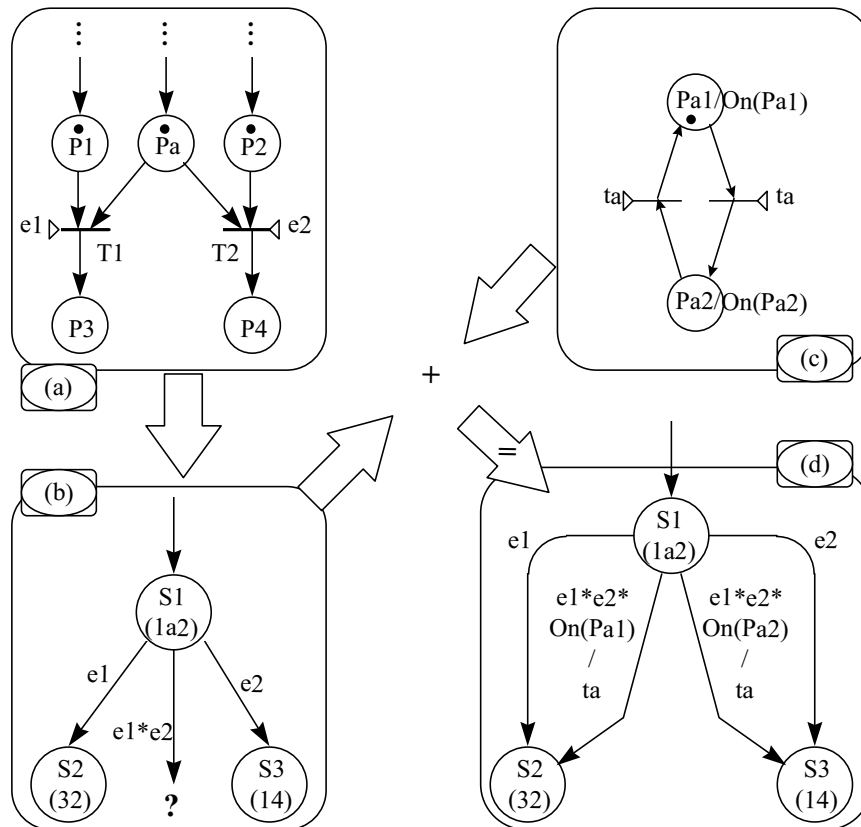


Figura 3-16 - Resolução de conflito: a) modelo inicial; b) espaço de estados associado incluindo não-determinismo; c) árbitro auxiliar; d) espaço de estados alterado.

Segue-se uma breve descrição dos procedimentos utilizados para a resolução dos conflitos. O processo inicia-se com a detecção de situações de conflito efectivos, ver Figura 3-16(b), marcando-se os arcos no espaço de estados inicial. Para cada arco marcado, é construído um árbitro (determinístico), ver Figura 3-16(c). O árbitro mais simples é composto por um anel de estados mono-marcado (isto é, uma máquina de estados), com tantos lugares quantas as possibilidades de resolução dos conflitos associadas a esse estado. As transições entre os lugares do anel construído têm um evento externo associado ( $ta$  na Figura 3-16(c)) e todos os lugares actuam uma saída específica quando marcados ( $On(Pa_1)$  e  $On(Pa_2)$  na Figura 3-16(c)). Cada arco marcado no espaço de estados inicial será decomposto em tantos arcos quantas as possibilidades de resolução dos conflitos; a cada um dos arcos assim gerados é associado uma condição externa resultante da composição da expressão associada ao conflito ( $e_1$  AND  $e_2$  no caso do exemplo) com uma das saídas produzidas pelo árbitro.

Finalmente, cada novo arco, aquando da mudança de estado, actua uma saída que provocará a alteração do estado do árbitro utilizado (ta no caso do exemplo).

Podem ser utilizados árbitros alternativos, como as já referidas máquinas de estado implementando probabilidades de disparo ou sistemas dedicados de regras, desde que as características gerais sejam mantidas.

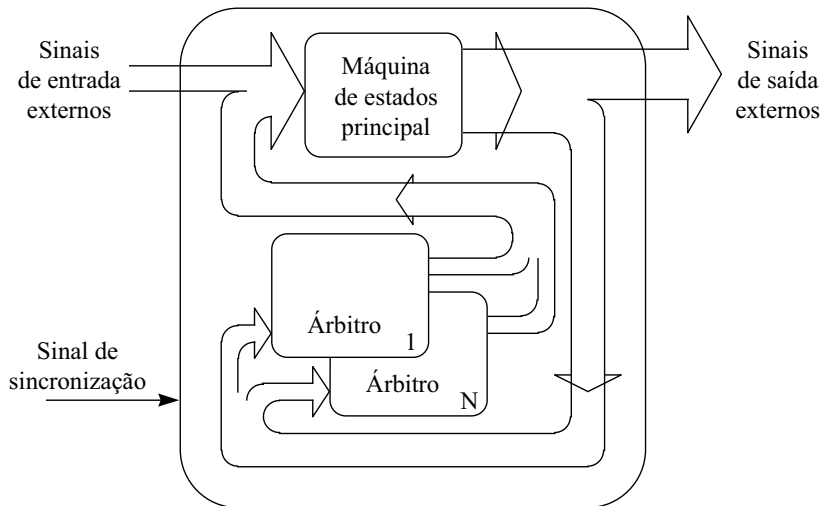


Figura 3-17 - Estrutura modular de blocos, aplicável na resolução de conflitos em implementações baseadas na execução do espaço de estados.

A execução do modelo inicial (Figura 3-16(a) no exemplo) é alcançada através da execução concomitante de  $N$  máquinas de estado, em que  $N$  é o número de conflitos efectivos identificados mais um (no exemplo, é necessário considerar duas máquinas de estado, apresentadas nas Figura 3-16(c) e Figura 3-16(d)), como ilustrado na Figura 3-17. Os sinais de entrada e saída referidos incluem quer sinais permanentes quer eventos.

Em termos gerais, a Figura 3-17 pode ser considerada como uma referência para implementações baseadas no espaço de estados. Dependendo das características seleccionadas para os árbitros, procede-se à geração da máquina de estados e o tipo de resolução de conflito fica determinado (equitativo, determinístico, etc).

### 3.4.1.c Resolução não-determinística e equitativa

A análise que se segue relaciona-se com as regiões II e VI da Figura 3-11, onde se pretende manter o não-determinismo em resoluções de conflito equitativas. O suporte ao não-determinismo é garantido através dos já propostos métodos utilizando árbitros. Neste caso, um árbitro típico será um gerador de números aleatórios (ou outro sistema autónomo externo funcionalmente semelhante). A análise é

iniciada pelas implementações baseadas na execução do espaço de estados, associadas à região VI da Figura 3-11.

Tomando a Figura 3-16 como referência, a solução proposta substitui o árbitro implementado através de uma máquina de estados (com capacidades de comunicar através de variáveis de entrada e saída com o exterior) por um gerador de números aleatórios (cuja saída é actualizada em todos os ciclos de tratamento).

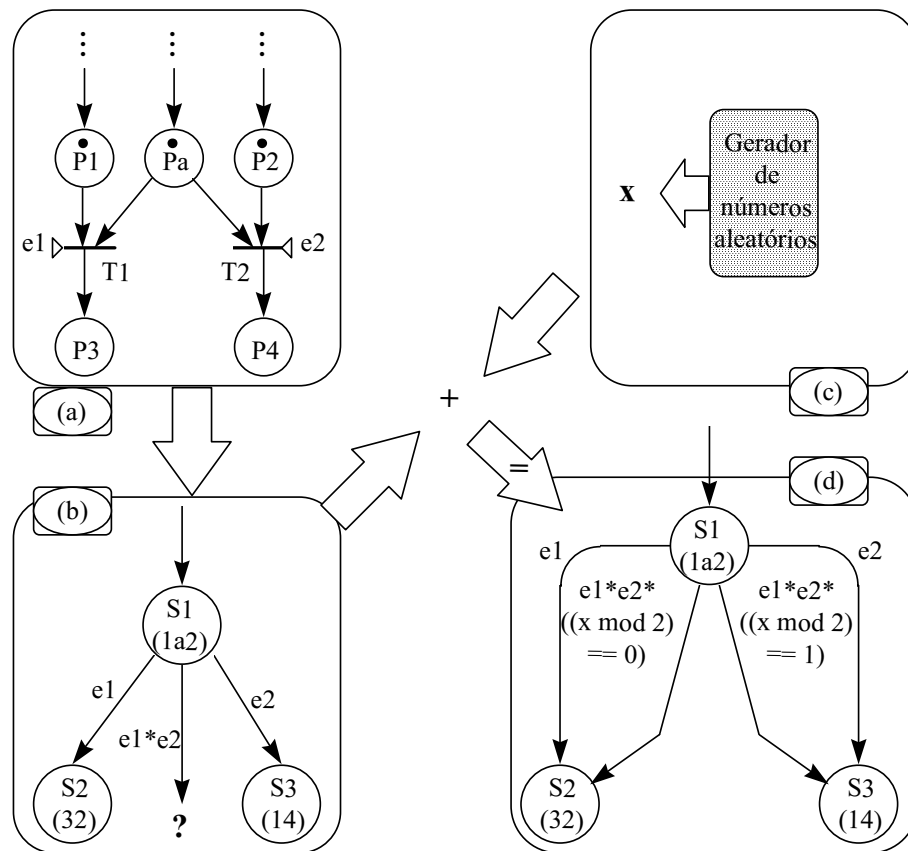


Figura 3-18 - Resolução de conflito: a) modelo inicial; b) espaço de estados associado incluindo não-determinismo; c) árbitro auxiliar; d) espaço de estados alterado.

O processo é apresentado na Figura 3-18, onde se retoma o exemplo do semáforo de exclusão mútua. O processo é semelhante ao da Figura 3-16. Neste caso, a resolução do conflito fica dependente da saída de um gerador aleatório, traduzida através das expressões de transição entre nós do grafo de estados final (ver Figura 3-18(d)). Significa isto que o não-determinismo é garantido não pelo grafo de estados obtido, mas por um agente externo ao modelo que condiciona o seu comportamento. Considerando que a saída do gerador é, implicitamente, actualizada em todos os ciclos de tratamento, deixa de ser necessário a geração de saídas pelo grafo de estados final.

A definição e implementação de “probabilidades de disparo” ou “equidade ponderada” é, novamente, trivial.

A execução do modelo inicial (Figura 3-18(a), no exemplo) é implementada através da execução concorrente de uma máquina de estados (Figura 3-18(d), no exemplo) e de um gerador aleatório.

Entre os benefícios desta solução, o mais significativo é a quantificação exacta dos recursos adicionais necessários para a implementação desta estratégia de resolução de conflitos. Neste caso, independentemente do número de conflitos existente, é necessário um único gerador de números aleatórios. Entre as desvantagens óbvias, conta-se a limitação da sua utilização em plataformas que disponham deste tipo de recurso. Um número significativo de controladores programáveis industriais são, desta forma, excluídos, podendo, alternativamente ser implementado um gerador pseudo-aleatório de comprimento pré-definido (muito pequeno), situação que se confunde com a solução proposta na secção anterior.

Tendo as implementações baseadas em jogadores-de-marcas em mente, também é possível embeber nas estratégias utilizadas para resolução de conflitos um gerador aleatório, utilizável por todo o modelo. Esta solução coloca-se na região II da Figura 3-11, embora baseada no que pode ser classificado como jogador-de-marcas estendido, isto é, um jogador-de-marcas em que nem toda a informação está nas marcas objecto do jogo, estando alguma escondida no próprio jogador.

## **3.4.2 Sistematização das soluções propostas**

### *3.4.2.a Carta de selecção de solução*

Como mencionado, pretende-se, no âmbito deste trabalho, utilizar as RdP não apenas como um formalismo que permita a especificação de controlo e que suporte a simulação e validação do modelo, mas que, para além disso, permita suportar todo o processo, desde a especificação à implementação. Um dos objectivos da metodologia é, como referido, obter, de forma automática, código executável que possa ser directamente embebido em sistemas.

Neste contexto, foram apresentadas várias estratégias aplicáveis no sentido de resolver automaticamente conflitos nas RdP-R. Estas estratégias podem ser embebidas no jogador-de-marcas ou no jogador-de-estados, ou indicarem a dependência em relação a sistemas externos cooperantes.

A Figura 3-19 apresenta uma carta de suporte à selecção entre as várias soluções propostas. A escolha de um determinado caminho nesta carta corresponde à selecção de uma determinada estratégia de resolução de conflitos.

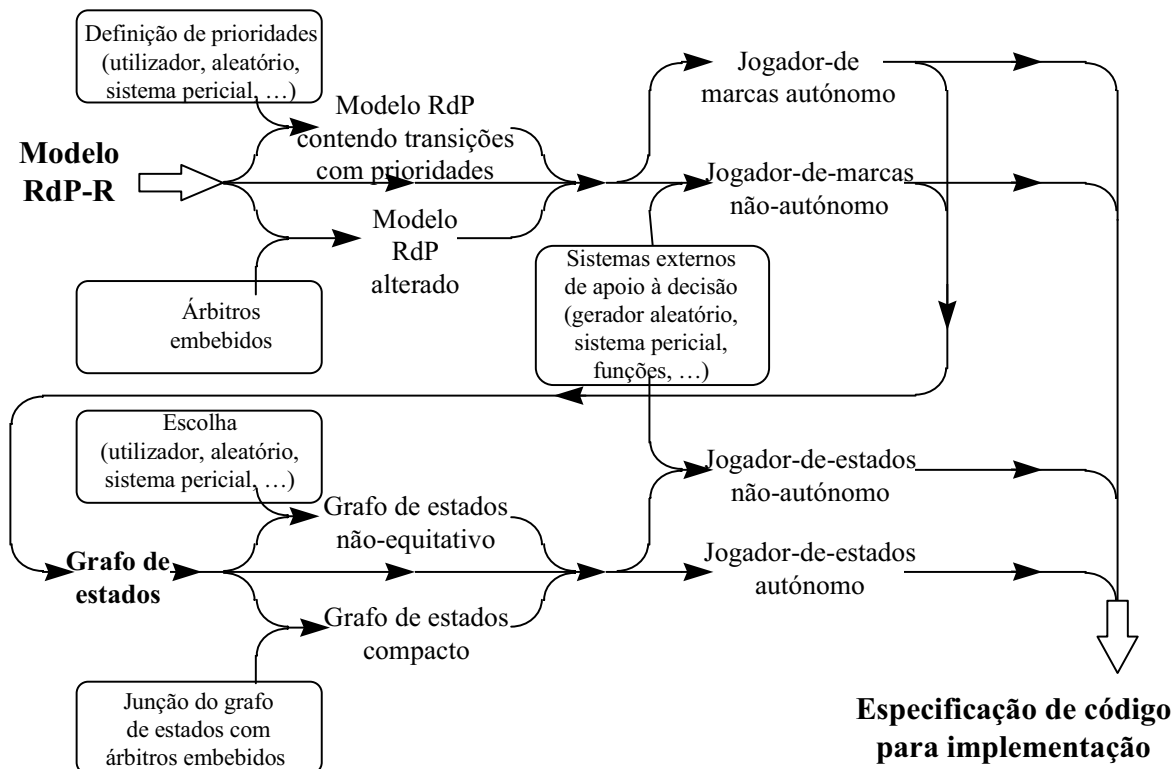


Figura 3-19 - Carta de soluções para a resolução de conflitos.

Na Figura 3-19 podem ser identificadas três regiões distintas:

- na parte superior e canto superior esquerdo, as soluções baseadas em jogadores-de-marcas;
- na parte inferior e canto inferior esquerdo, as soluções baseadas em jogadores-de-estados;
- na parte central à direita, as soluções não-autónomas, isto é, dependentes de sistemas externos de apoio à decisão a serem utilizados em tempo de execução.

O código a ser utilizado para efeitos de implementação pode ser obtido a partir de um dos quatro jogadores apresentados:

- jogadores autónomos, que leem os sinais de entrada e determinam o próximo estado, eventos e ações de saída com base no modelo (e suas características intrínsecas) ou no grafo de estados;
- jogadores não-autónomos, que juntam à execução da RdP-R ou do grafo de estados associado, as “sugestões” produzidas por sistemas externos de apoio à decisão. O sistema de apoio à decisão pode ser tão simples como um gerador aleatório ou tão complexo como (alguns) sistemas periciais. A utilização de sistemas periciais difusos revela-se de elevado interesse para a resolução de conflitos em situações complexas; como exemplo, refira-se

que a utilização equilibrada de recursos numa planta fabril pode ser gerida por um sistema desse tipo. Neste caso, a utilização de lógica difusa apresenta-se adequada à modelação de algumas acções, como abordado noutra capítulo deste trabalho, e a tradução do conjunto de regras difusas em RdP-R (como proposto em [Gomes & Steiger-Garção, 95d]) permite a validação do modelo global.

#### *3.4.2.b Sobre a indicação do método de resolução de conflitos*

A implementação das soluções para resolução de conflitos apresentadas nas secções anteriores é suportada directamente pelas características identificadas no capítulo anterior para as RdP-R, sendo necessário associar a cada transição envolvida em conflitos dois tipos de informação:

- a primeira está relacionada com o tipo de mecanismo de resolução de conflito seleccionado (de entre as soluções propostas);
- a segunda informação é um parâmetro específico da solução seleccionada. Nalguns casos esta informação refere-se à prioridade, enquanto noutros armazena a probabilidade de disparo (ou equidade ponderada); noutros ainda, não é necessária.

## 4. As Redes de Petri Reactivas e Hierárquicas

*A união faz a força  
(Provérbio popular)*

### *Sumário do capítulo*

As capacidades de estruturação de um modelo são essenciais na especificação de sistemas. A tradicional atitude de “dividir para reinar”, procurando o reverso do provérbio que abre este capítulo, é seguida em inúmeras metodologias. É, no entanto insuficiente quando se consideram os sistemas reactivos de tempo-real. No presente capítulo serão propostos mecanismos de decomposição e representação hierárquica específicos, dando origem às RdP Reactivas e Hierárquicas (RdP-RH). Assim, este capítulo é composto pelas seguintes secções:

#### *4.1 Introdução*

#### *4.2 Representações hierárquicas e modulares com RdP*

#### *4.3 Propostas para a modelação hierárquica e modular de sistemas reactivos de tempo-real*

#### *4.4 Exemplo de aplicação*

Na primeira secção procede-se a um levantamento das atitudes comuns nas abordagens à modelação de sistemas de média e elevada complexidade. Os “reflexos” verificados na área das RdP serão apresentados na parte final da secção.

Um levantamento detalhado das propostas e soluções existentes de extensões e classes de RdP vocacionadas para o tratamento da complexidade do modelo será realizado na segunda secção.

As propostas de mecanismos específicos de decomposição e representação hierárquica serão apresentadas na terceira secção. Propõem-se três mecanismos básicos de estruturação do modelo, denominados por decomposição horizontal, vertical e mista, respectivamente. Associados a cada um deles serão definidos novos tipos de nós, denominados por macronós, metanós e supernós, respectivamente. Acessoriamente, propõe-se a representação estruturada baseada em vectores

associada a qualquer nó da rede. Será demonstrada a possibilidade de encontrar modelos RdP-R comportamentalmente equivalentes aos modelos RdP-RH propostos. No sentido de ilustrar a aplicabilidade das técnicas propostas, serão apresentados os procedimentos que permitem traduzir modelos statechart em RdP-RH comportamentalmente equivalentes.

Na quarta secção procede-se à apresentação de um exemplo de aplicação típico (um sistema produtor-consumidor) em torno do qual será possível ilustrar a aplicação dos vários mecanismos de modelação propostos.

## 3. As Redes de Petri Reactivas e Hierárquicas

### 4.1 Introdução

Considerando a modelação de sistemas de média ou elevada complexidade, a utilização de mecanismos de estruturação do modelo que permitam a subida dos níveis de abstracção utilizados é uma necessidade incontornável. Inicia-se o presente capítulo com a apresentação de um conjunto de motivações e soluções encontradas ao longo dos últimos anos para tratar do problema da gestão da complexidade de modelos de sistemas.

De acordo com a área de aplicação ou o objectivo da actividade de modelação, ênfases diferentes podem ser colocadas em vários aspectos; a título de exemplo, referem-se as seguintes motivações:

- necessidade de vários níveis de abstracção para a especificação e análise do modelo,
- necessidade de verificar o modelo obtido, sendo a análise do modelo o motor de desenvolvimento,
- necessidade de validação do modelo, através de simulação,
- necessidade de descrição compacta de sistemas.

Tomem-se, como referências de motivação para o tema, os exemplos associados aos métodos de desenvolvimento de sistemas computacionais, nas suas componentes hardware e software, separadas como se mantinham alguns anos atrás ou integradas através de metodologias de *co-design*, como encaradas actualmente.

A utilização dos circuitos electrónicos digitais, independentemente das facilidades e limitações que as tecnologias possam, em determinada altura, ter tornado disponíveis, tem vindo a beneficiar da disponibilidade de uma biblioteca vasta e diversificada de circuitos integrados com funções específicas. A evolução transistor, porta lógica, elemento de memória, funções combinatórias, sistemas sequenciais específicos ilustra de forma clara a capacidade de disponibilizar módulos que convenientemente combinados, suportam de forma eficaz o desenvolvimento de sistemas complexos.

O conceito de modularidade constitui-se, deste modo, como uma atitude básica nas actividades de desenvolvimento de sistemas electrónicos.

Às bibliotecas (imensas) de módulos disponíveis comercialmente (isto é, os vulgares circuitos integrados), juntaram-se-lhe as bibliotecas que cada utilizador pode construir através dos dispositivos de lógica programável, normalmente referidos por PLDs (de “Programmable Logic Devices”). Neles se incluem desde os dispositivos de reduzida complexidade, como as PALs (de “Programmable Array Logic), aos dispositivos de elevada complexidade, como as FPGA (de “Field-Programmable Gate Array”), passando pelas complexidades intermédias, como os CPLDs (de “Complex Programmable Logic Devices”).

A atitude de utilização de módulos mantém-se quando se abandona o projecto com componentes discretos e se considera a utilização de circuitos integrados de aplicação específica (comumente referidos por ASICs, de “Application Specific Integrated Circuit”), em cujos ambientes de desenvolvimento o objectivo é, normalmente, de especificar e produzir circuitos autónomos de elevada complexidade.

Deste modo, na área de desenvolvimento de hardware, as vantagens características de abordagens que potenciam caracterizações modulares são, assim, de menção obrigatória.

Na área da programação de sistemas, tentando também referir apenas alguns pontos notáveis, à “era” da programação em “assembly” (em que a utilização de macros se apresentava como algo “sofisticada” e efectivamente muito útil), seguiram-se, naturalmente, outras, como a das linguagens de alto nível (caracterização actualmente pouco esclarecedora), da programação estruturada, por eventos e orientada por objectos, apenas para referir algumas. A construção de bibliotecas (de sub-rotinas, funções ou classes, dependendo dos paradigmas utilizados) constitui-se como o equivalente do conceito de “circuito integrado”, nesta área de desenvolvimento.

Uma outra atitude de projecto, fortemente cooperante com a abordagem modular, pode caracterizar-se pela utilização de uma estruturação do sistema ou dos conceitos envolvidos de modo hierárquico. A utilização deste paradigma conduz à caracterização do sistema em vários níveis de abstracção, podendo a caracterização incidir sobre os conceitos tratados em cada um dos níveis ou sobre a decomposição em vários subsistemas através de (vários) refinamentos sucessivos.

Em ambas as áreas apresentadas (*software e hardware*), fica claro que os projectistas envolvidos têm vindo a dispôr de técnicas cada vez mais potentes para o desenvolvimento de sistemas, acompanhadas da disponibilidade de ambientes computacionais de apoio ao desenvolvimento de complexidade

crescente. Estes ambientes comportam-se como ferramentas de análise e implementação do projecto, permitindo a detecção e correcção antecipadas de eventuais erros.

A conjugação das abordagens referidas tem permitido a reutilização de resultados anteriormente obtidos, nomeadamente através da sua integração em novos projectos. O rendimento assim obtido, bem como a qualidade dos sistemas desenvolvidos, aumenta significativamente, dado que os vários subsistemas beneficiam da acumulação dos testes de qualidade anteriormente realizados.

As idéias de modularidade e de decomposição e agregação são extensivamente utilizadas nas abordagens de análise de sistemas comumente classificadas em abordagens descendente (“*top-down*”) e ascendente (“*bottom-up*”).

Nas abordagens descendentes, cada um dos subsistemas, obtidos a um qualquer nível da representação, poderá ser alvo de um refinamento posterior, aplicado iterativamente, até que os submodelos obtidos sejam suficientemente simples para se proceder à sua análise ou implementação directas.

Nas abordagens ascendentes, a utilização e combinação de blocos previamente disponíveis permitirá obter o comportamento desejado para o (sub)sistema.

Concomitantemente poderão ser utilizadas abordagens estruturadas que permitem compactar os modelos utilizados através de mecanismos de compaginação (“*folding*”), normalmente suportados por características de repetição ou de simetria presentes nos modelos. A título de exemplo, refira-se a possibilidade de representar um sistema constituído por  $N$  subsistemas idênticos, por um único modelo representativo de um subsistema ao qual é adicionada uma característica de repetição por  $N$  [Ohr, 90]. Neste caso, a escolha da classificação como “estruturada” fica naturalmente associada à utilização de estruturas de dados para suportar a compactação. Em [Gomes, 91: pp.33] é apresentado um exemplo significativo, aplicado à especificação gráfica compacta de circuitos digitais, em que a representação compacta de um conjunto de  $N$  módulos de tipo  $X$  é realizada através de um “vector” de módulos,  $X[N]$ .

No contexto das redes de Petri, de referir as RdP Coloridas [Jensen, 92] em que o mecanismo de paginação é extensivamente utilizado.

#### **4.1.1 Relações com os modelos de RdP**

No caso particular da evolução dos modelos baseados em RdP, podem ser identificadas várias contribuições em sintonia com as tendências apresentadas em relação às metodologias de especificação, análise e realização de sistemas computacionais.

Tentando identificar algumas referências importantes na evolução da teoria e aplicação das RdP verifica-se que, justificado pelo sólido suporte teórico associado às RdP de baixo nível propostas nos anos 60 e desenvolvido nos anos 70, foram realizadas e aprofundadas várias propostas de RdP de alto-nível durante os anos 80. As propostas de RdP hierárquicas e modulares aparecem e são aprofundadas durante os anos 90. Recentemente algumas propostas têm vindo a integrar o paradigma da orientação por objectos em modelos de RdP.

De entre as “graves” incapacidades normalmente apontadas às RdP de baixo nível refiram-se [He & Lee, 91]:

- a estrutura rígida da RdP transforma qualquer mudança a um modelo já existente numa tarefa de difícil execução, nomeadamente em termos da verificação de propriedades no modelo resultante;
- a impossibilidade de representar marcas individualizadas, isto é, com informação particular;
- devido à ausência de técnicas de estruturação hierárquica do modelo, as especificações resultantes são, normalmente demasiado extensas; a imagem do grafo com as dimensões de um campo de futebol é, demasiadas vezes, utilizada como metáfora devastadora.

Característica importante, do ponto de vista da solidez e aceitação de um formalismo, é a capacidade das propostas de novas classes serem acompanhadas com trabalhos que as relacionam com classes já conhecidas. Desta forma, permite-se a sua tradução mútua e obtenção de representações equivalentes (do ponto de vista comportamental ou das propriedades associadas, para referir algumas das características de interesse). Esta é a situação em classes de RdP existentes que promovem a compactação e/ou a representação hierárquica, para as quais é possível aplicar resultados já obtidos para RdP menos complexas.

Como exemplos paradigmáticos de um conjunto de trabalhos satisfazendo as características indicadas no tocante a modularidade e hierarquia, refiram-se as RdP Coloridas Hierárquicas [Jensen, 92] [Jensen, 95]. As RdP Coloridas Hierárquicas podem ser traduzidas em termos de RdP Coloridas; por sua vez, as RdP Coloridas podem ser caracterizadas em termos de RdP Lugar-Transição. Deste modo, os resultados aplicáveis a RdP Lugar-Transição podem ser reutilizáveis nas RdP Coloridas e Hierárquicas.

Obviamente que, do ponto de vista de engenharia, o facto das várias classes referidas disporem de capacidades semelhantes de modelação, não retira interesse à utilização de modelos mais compactos,

embora, eventualmente, menos intuitivos, dado que a extensão do modelo é condensada através da utilização de anotações da rede.

## **4.2 Representações hierárquicas e modulares com RdP**

Nos formalismos de descrição de sistemas em que a notação utilizada recorre a uma representação gráfica, como é o caso das RdP, as questões associadas à capacidade de produzir representações compactas revestem-se de primordial importância. Doutra forma a dificuldade em apreender o comportamento de um modelo extenso, bem como a necessidade de utilizar folhas de dimensões não usuais podem constituir-se em fortes desmotivações à adopção da metodologia. Como referido, a capacidade de produção de especificações compactas é suportada através da utilização cooperante de técnicas estruturadas, modulares e de decomposição/agregação hierárquicas.

Na presente secção são referidos os trabalhos considerados mais relevantes utilizando representações hierárquicas e modulares de RdP, bem como aspectos ligados à notação gráfica utilizada, com interesse para a proposta a apresentar. Inicia-se por um levantamento do estado da arte e conclui-se com alguns comentários identificando temas necessitando de evolução, efectuando o enquadramento das propostas a apresentar.

Para a caracterização do estado da arte foi escolhido um critério de apresentação cronológico, dado corresponder simultaneamente e grosso modo à evolução da teoria e aplicações de RdP. Assim, inicia-se com a caracterização das técnicas utilizadas em RdP de baixo-nível, seguindo-se as RdP de alto-nível, nomeadamente as RdP Coloridas Hierárquicas, as RdP Predicado-Transição Hierárquicas e alguns modelos orientados por objectos, com especial ênfase para as RdP Objecto.

Nesta ordem de apresentação é claro um aumento da importância da notação gráfica utilizada, tornando-a mais sofisticada. Uma outra característica com importância acrescida neste percurso é a utilização extensiva de inscrições na RdP, nomeadamente nas classes de RdP de alto-nível. Em certo sentido, as estruturas de dados e a algoritmia associadas à modelação do sistema têm vindo a ser embebidas no modelo através da utilização de inscrições, substituindo ou “escondendo” a representação gráfica das RdP de baixo-nível (com base na qual é possível inferir sobre algumas características comportamentais de forma mais intuitiva). Este procedimento, presente nalgumas propostas que utilizam RdP embebidas em metodologias orientadas por objectos, conduz, muitas vezes, a modelos de difícil formalização e para os quais as técnicas de análise disponíveis são reduzidas ou inexistentes.



Figura 4-1 - Modelação de um evento não-primitivo.

#### 4.2.1 Utilizando RdP de baixo-nível

O interesse em utilizar técnicas de refinamento na modelação através de RdP foi identificado desde cedo. Dado que os eventos considerados nas RdP são instantâneos, a modelação de eventos com duração não-nula (denominados por eventos não-primitivos) foi resolvida através da utilização de dois eventos instantâneos (denominados eventos primitivos), modelando o início e fim do evento não-primitivo, e de um lugar modelando a sua ocorrência (ver Figura 4-1).

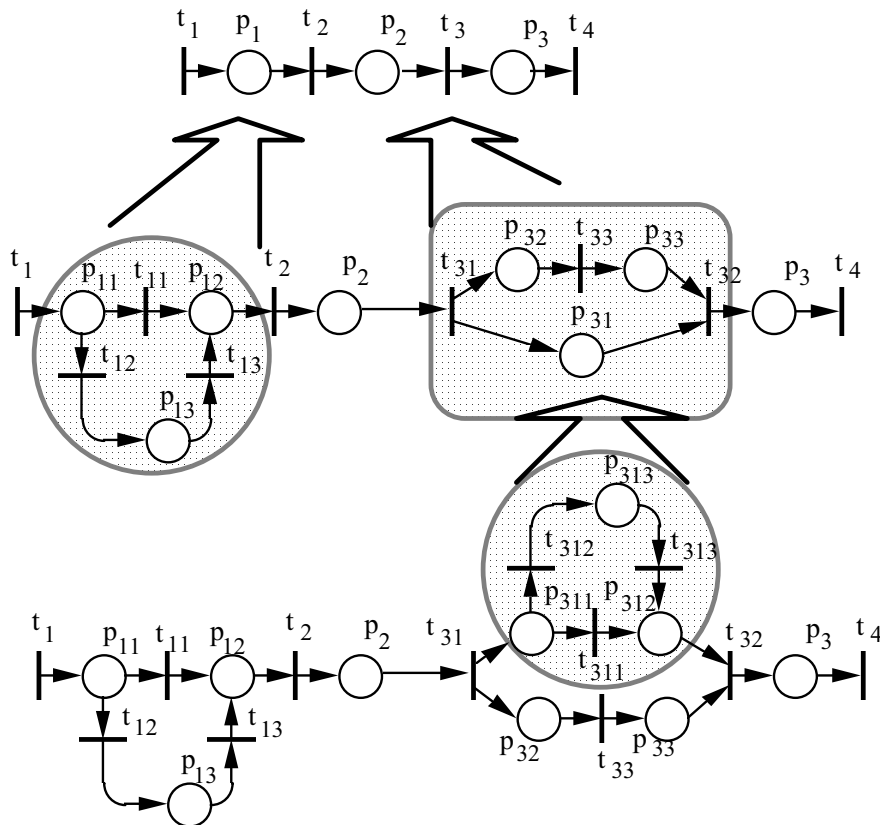


Figura 4-2 - Modelação hierárquica utilizando refinamentos sucessivos.

Alguns investigadores, incluindo C. A. Petri, sugeriram que os eventos não-primitivos fossem representados por rectângulos. Introduz-se, assim, uma representação distinta para transições que podem sofrer refinamentos posteriores.

Este tipo de procedimento é semelhante ao utilizado nas macros em linguagem *assembly*; por analogia, o termo macrotransição é, normalmente, utilizado para representar este tipo de decomposição hierárquica em que uma sub-RdP é utilizada para substituir uma transição.

Esta "macro-representação" de uma sub-RdP por uma transição pode ser estendida aos lugares, denominados macrolugares, tornando-os também numa representação de uma sub-rede. A decomposição do modelo pode ser realizada em vários níveis da representação, através de refinamentos sucessivos, como se apresenta na Figura 4-2.

Em [Fehling, 93] são definidos conceitos de RdP Hierárquicas, no sentido de suporte à modelação descendente (em linha com o que se apresentou nos parágrafos precedentes), e de RdP com Blocos ("with Building Blocks"), no sentido do suporte à modelação modular ascendente. Os Blocos são caracterizados como módulos, podendo ser utilizados com diferentes objectivos, nomeadamente:

- reutilização de redes;
- armazenamento de apenas uma instância de uma rede utilizada múltiplas vezes;
- chamada dinâmica de sub-redes.

As questões associadas à chamada dinâmica de sub-redes não são precisadas, sendo as propostas contidas em [Huber, Jensen & Shapiro, 90] utilizadas como referência (este trabalho será apresentado mais à frente neste capítulo, quando da abordagem das RdP Coloridas).

Obviamente que, tomando como referência as linguagens de programação, o passo seguinte para a estruturação de um modelo de RdP será o de re-utilizar submodelos numa disciplina tipo subprograma (em lugar da disciplina tipo "macro").

Considere-se como exemplo na área de modelação do controlo de sistemas, o caso elementar da representação de uma sequência de controlo utilizável a partir de várias situações. Pode ser entendida como o caso mais simples de subprograma, em que este não dispõe de argumentos, nem de entrada, nem de saída. A Figura 4-3 propõe um esquema básico para a sua modelação [Silva, 85: 45] para o caso de RdP seguras em que não seja possível a utilização concomitante do subprograma a partir de dois ou mais pontos de chamada (isto é, não suporta a modelação de procedimentos com reentrância); os lugares  $p_i$  e  $p_j$  memorizam o local de chamada da actividade SP, permitindo determinar o ponto de regresso após conclusão.

Em subsecção posterior deste capítulo, aquando da análise das RdP Coloridas, a modelação de sub-rotinas será retomada através da análise das propostas contidas em [Huber, Jensen & Shapiro, 90], nomeadamente do mecanismo denominado por evocação de transições.

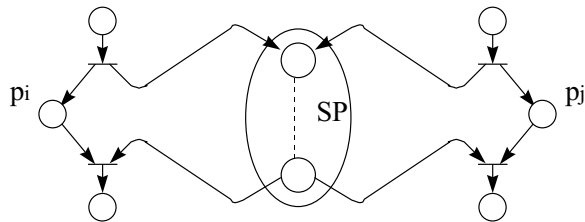


Figura 4-3 - Esquema básico para modelação de subprogramas.

Tomem-se como referência as classes de RdP de baixo-nível em que os lugares podem conter o máximo de uma marca, onde se incluem as RdP Condição-Evento, as RdP Elementares e as RdP Seguras. Podem ser identificados diversos trabalhos em que o objectivo é o de desenvolver métodos de composição de modelos, de alguma forma classificáveis como atitudes de modelação modulares. O trabalho apresentado em [Bernardinello & De Cindio, 92] constitui uma excelente referência, sistematizando diversas abordagens propostas.

Grande parte desses trabalhos partilham características comuns, nomeadamente a de tentarem construir o modelo do sistema através da composição das suas componentes sequenciais, representadas através de máquinas de estado. Numa máquina de estados podem ser modeladas situações de conflito, mas não de concorrência; a concorrência pode ser introduzida quando várias máquinas de estado são consideradas conjuntamente para modelar o sistema.

Uma das técnicas possíveis para garantir a comunicação entre as várias máquinas de estado é a de sobreposição de transições partilhadas por várias máquinas de estado, permitindo implementar a evolução síncrona das máquinas de estado envolvidas. Como exemplo, refira-se o modelo SA (Superposed Automata), referido em [Bernardinello & De Cindio, 92] e integrado em ambientes de especificação com RdP de alto-nível a referir mais à frente [Battiston & Cindio, 93]; aqui a composição das várias máquinas de estado é conseguida através da utilização de um operador denominado “composição-T” em que a habilitação de todas as transições homónimas presentes nas várias máquinas de estado é avaliada para determinar a habilitação da transição partilhada.

Na Figura 4-4 apresenta-se um exemplo típico de modelação de exclusão mútua em relação a um recurso comum (retirado de [Bernardinello & De Cindio, 92]); a RdP N representa o modelo global, enquanto as RdP N1, N2 e N3 modelam as máquinas de estado elementares sincronizadas através das transições A, B, C e D.

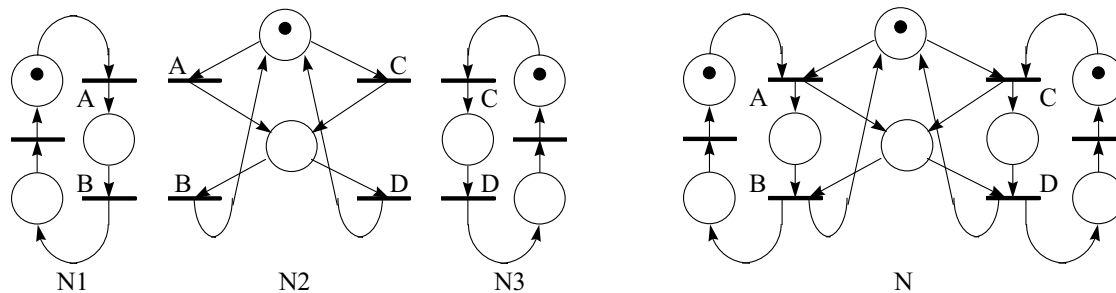


Figura 4-4 - Um modelo SA (N) com as suas três máquinas de estado elementares (N1, N2 e N3) modelando uma exclusão mútua em relação a um recurso comum.

Considerando a classe das RdP Lugar-Transição, existem na literatura referências para um vasto trabalho na área de caracterização modular e hierárquica de RdP. Como exemplo, refira-se o modelo  $\{SC\}^*ECS$  (Systems of Cooperating Systems of ... Cooperating Equal Conflict Systems) [Recalde, Teruel & Silva, 96], em que se utiliza um mecanismo de comunicação assíncrona entre os vários módulos, implementados através de “*buffers*”, aplicável a uma classe específica de RdP Seguras.

É interessante verificar que algumas das técnicas de estruturação do modelo apresentadas como utilizáveis em tempo de especificação do sistema, são encontradas também quando o objectivo é o de analisar ou simplificar o modelo original através de técnicas de redução, mantendo um conjunto de propriedades dinâmicas.

Neste pressuposto, a utilização de técnicas de estruturação do modelo, resultado de uma abordagem específica escolhida pelo projectista, resulta numa identificação a priori das situações de simplificação aplicáveis. A identificação de situações de possível simplificação é tarefa que, a ser consequência de uma atitude associada às abordagens ascendentes (isto é, partindo do modelo plano), necessita, no caso geral, de um forte suporte algorítmico no sentido da sua identificação (no caso dos modelos fortemente compactados, como é o caso dos obtidos com RdP Coloridas Hierárquicas, esta identificação apresenta-se como de elevada complexidade).

Deste modo, é interessante enfatizar a interdependência existente entre o processo de análise de uma RdP utilizando técnicas de redução, o processo de descrição de um modelo utilizando refinamentos sucessivos (abordagem descendente) e o processo de simplificação visando a implementação ou a análise utilizando métodos de simplificação. Esta forte relação entre técnicas de redução e simplificação e de refinamento permite que resultados de cada uma das técnicas possam ser aplicados nas outras.

Como exemplo ilustrativo desta relação ao nível da terminologia utilizada, refira-se que o termo *macrolugar*, já introduzido nesta secção como uma técnica de estruturação do modelo por refinamentos sucessivos, e utilizado em imensos trabalhos para significar *substituição de lugares* (no

sentido utilizado para as RdP de alto-nível em [Huber, Jensen & Shapiro, 90]), é introduzido em [Silva, 85] no âmbito dos métodos de análise por redução de RdP Ordinárias. Uma definição formal de macrolugar é realizada em [Silva, 85: 112], incluindo-se algoritmos para a sua obtenção, permitindo a redução de uma sub-rede a um lugar, impondo um conjunto de restrições estruturais na sub-rede reduzida e tendo associado uma marcação que é simplesmente a soma de todas as marcas presentes na sub-rede. Com a caracterização referida é possível identificar um conjunto de propriedades que são preservadas após o processo de redução por macrolugares, nomeadamente se a RdP inicial é viva e limitada, a RdP reduzida também o é.

Enfatizando esta interdependência forte e explícita entre métodos de análise, técnicas de simplificação e de estruturação do modelo de RdP, em [Silva, 85], tendo por objectivo a simplificação de uma especificação dada, tenta-se reduzir (e não minimizar) o número de lugares e transições do modelo inicial, conservando o “sentido físico” da descrição inicial, através de simplificações locais. A simplificação do modelo visa a sua redução efectiva (em número de lugares e transições), podendo ser utilizada concomitantemente com os métodos de análise por redução.

Os tipos de simplificação considerados em [Silva, 85: 64] são classificados em dois grupos:

- simplificações estruturais, em que se pretende eliminar redundâncias de tipo estrutural, existentes na definição de falsas evoluções paralelas; dependem da estrutura e da marcação inicial da RdP e baseiam-se na eliminação de lugares implícitos;
- simplificações que têm em conta a interpretação associada à RdP, isto é, a componente não-autónoma do modelo; são consideradas duas técnicas:
  - fusão de lugares, permitindo simplificar a parte sequencial da especificação. A fusão de lugares é realizada após a detecção dos lugares equivalentes e compatíveis (classificação que é realizada tendo em conta a interpretação associada à RdP);
  - supressão de ligações entre lugares, com eliminação de lugares, através da aplicação da técnica dos lugares fonte.

As simplificações estruturais propostas em [Silva, 85] baseiam-se na identificação e remoção de lugares implícitos. O conceito de lugar implícito foi proposto originalmente na tese de G. Berthlot [Berthlot, 83] (ver também [Berthlot, 85]). De uma forma intuitiva, a marcação de um lugar implícito pode ser determinada por uma função dependente do estado de marcação de um conjunto de outros lugares; adicionalmente, nunca um lugar implícito será o único lugar ligado às suas transições de

saída. Em [Silva, 85: 65] apresentam-se técnicas de detecção e eliminação de lugares implícitos aplicáveis a RdP Seguras.

Na Figura 4-5 apresenta-se um exemplo de uma RdP com um lugar implícito (p2); considerando que p3 e p4 podem ser representados por um macrolugar, fica clara a redundância entre p2 e o macrolugar.

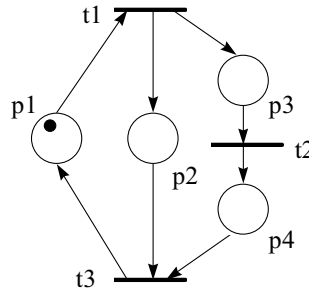


Figura 4-5 - p2 é um lugar implícito; está “em paralelo” com p3 e p4.

Quando ao lugar implícito não estejam associadas ações (no sentido dado nas RdP Interpretadas, assim como nas RdP Reactivas), então o lugar pode ser removido, sem consequências; caso existam ações associadas, deverão estas ser associadas a outros lugares de modo a permitir a remoção do lugar implícito. Claro que, do ponto de vista da implementação e embora permitindo reduzir o número de lugares, nem sempre a eliminação é aconselhada, dado que poderá corresponder a um acréscimo significativo de lógica combinatória (ou computação de expressões) associado à transferência das ações associadas ao lugar implícito para um conjunto de lugares.

Importante referir que a remoção de um lugar implícito não altera o comportamento do modelo, isto é, as sequências de disparo possíveis não se alteram com a remoção do lugar implícito. Na Figura 4-6 evidencia-se a dependência da marcação inicial na identificação de um lugar implícito [Silva, 96].

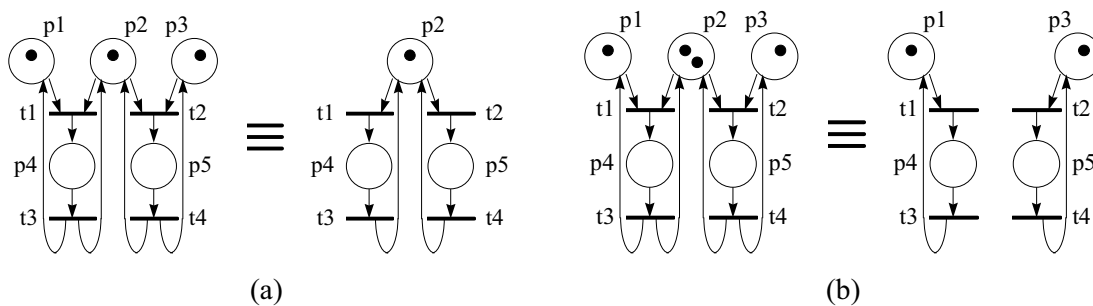


Figura 4-6 - (a) p1 e p3 são lugares implícitos; (b) p2 é lugar implícito.

Esta técnica de redução de lugares implícitos (para além de outras também contidas no trabalho inicial de Berthlot), pode ser estendida às RdP Coloridas, como realizado em [Haddad, 90].

## 4.2.2 Utilizando RdP de alto-nível

Do apresentado na secção anterior, pode concluir-se, provisoriamente, que as técnicas utilizadas para gerir a complexidade do modelo são extremamente diversificadas, de modo a suportar as tradicionais abordagens de especificação ascendente e descendente, utilizando mecanismos de decomposição modular e comunicação ou sincronismo entre módulos.

Claro que a utilização das técnicas apresentadas na secção anterior pode ser realizada concomitantemente com a utilização de RdP de alto-nível, aumentando as capacidades de compactação do modelo possíveis com a utilização de estruturas de dados associadas às marcas.

Algumas propostas significativas foram realizadas considerando uma classe específica de RdP de alto-nível (embora possam, eventualmente, ser utilizáveis nas outras classes), outras foram apresentadas de forma mais descomprometida.

De entre estas últimas, de referir a proposta de RdP de alto-nível Hierárquicas, HHPN (de “Hierarchical High level Petri nets”) [Buchholz, 94] vocacionadas para a análise de sistemas, tendo como motivação inicial a análise de cadeias de Markov. A especificação é decomposta em várias sub-redes (módulos) caracterizáveis por um interface e por uma marcação que especifica uma “vista” agregada da marcação interna da sub-rede (que corresponde à “vista” detalhada). O interface associado à vista agregada da sub-rede é realizado por um conjunto de portos de entrada e de saída que são embebidos na rede superior. Os portos de entrada e de saída podem ser divididos em lugares e transições-fronteira. Os portos dos lugares-fronteira podem ser ligados a transições ou a portos das transições-fronteira, enquanto que portos das transições-fronteira podem ser ligados a lugares ou a portos dos lugares-fronteira. Em cada sub-rede poderão existir um máximo de um porto de cada tipo (lugar de entrada, transição de entrada, lugar de saída, transição de saída). Em termos da representação gráfica utilizada, uma sub-rede será representada pela sua vista agregada composta por uma caixa contendo um lugar (eventualmente marcado) e os portos. Na Figura 4-7 apresenta-se um exemplo (abstracto) retirado de [Buchholz, 94].

Em contraste com a maioria das restantes propostas apresentadas, cada sub-rede é executável autonomamente, sendo o seu comportamento dinâmico descrito por um grafo de ocorrências local. As propostas presentes em [Buchholz, 94] mantêm-se sempre a um nível muito formal, sem apresentação de exemplos, pelo que a análise do impacto da metodologia em situações concretas reais fica em aberto.

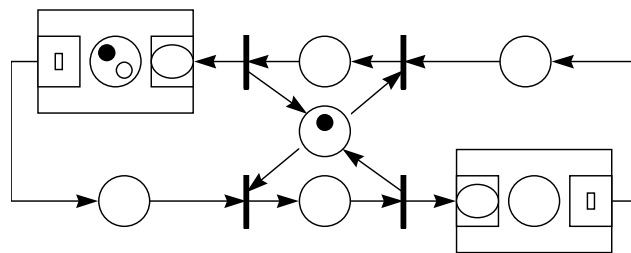


Figura 4-7 - Exemplo de uma rede HHPN contendo outras duas sub-redes HHPN.

Seguidamente, alguns trabalhos são analisados enquadrados dentro da classe em que foram propostos.

#### 4.2.2.a Utilizando RdP Coloridas

As RdP Coloridas são, sem dúvida, a classe de RdP que recebeu mais tempo de desenvolvimento em ferramentas computacionais para apoio à sua utilização. Não surpreende, portanto, que uma parte significativa dos trabalhos de interesse a referir utilizem RdP Coloridas.

A compactação de modelos através de RdP Coloridas, particularmente eficaz para modelos em que se verifiquem repetições ou simetrias, é suportada pela utilização de estruturas de dados associadas às marcas, permitindo a paginação (“*folding*”) do modelo.

Para além das RdP Coloridas representarem, em si, um progresso significativo no sentido da compactação de modelos, em [Huber, Jensen & Shapiro, 90] são propostos cinco mecanismos de representação hierárquica de modelos de RdP Coloridas. São eles:

- ⇒ substituição de transições,
- ⇒ substituição de lugares,
- ⇒ evocação de transições,
- ⇒ fusão de lugares e
- ⇒ fusão de transições.

Como se apresentará, alguns desses mecanismos permitem uma representação modular e/ou hierárquica do modelo, enquanto que outros podem ser confundidos com “meros” expedientes gráficos de apoio à produção de especificações mais “limpas”.

Sendo [Huber, Jensen & Shapiro, 90] um trabalho prévio de especificação do que mais tarde viria a ser formalizado nos livros sobre RdP Coloridas de Kurt Jensen [Jensen, 92] [Jensen, 95] e integrado no ambiente Design/CPN, é o trabalho com mais implicações na estruturação hierárquica de RdP

Coloridas e viu dois dos mecanismos propostos (substituição de transições e fusão de lugares) escolhidos como suficientes para suportar a definição e implementação das RdP Coloridas Hierárquicas [Jensen, 92].

As propostas apresentadas em [Huber, Jensen & Shapiro, 90] utilizam extensivamente o conceito de página, como armazenando uma sub-RdP Colorida. Cada página pode ser evocada qualquer número de vezes; a recursividade é permitida quando o mecanismo a utilizar é o de evocação de transições, sendo interdita na situação de substituição de transições ou lugares. Deste modo, suporta-se uma especificação modular do sistema em que a estrutura de dependência entre as páginas não é hierárquica, no sentido estrito do termo, podendo ser descrita como uma estrutura matricial onde se estabelecem relações hierárquicas específicas e onde as diversas páginas podem ser utilizadas múltiplas vezes (como módulos).

Devido à relevância do trabalho, caracterizam-se seguidamente e de forma breve os cinco mecanismos referidos.

#### 4.2.2.a.1 Substituição de transições

O primeiro mecanismo referido, de substituição de transições, é apresentado em [Huber, Jensen & Shapiro, 90] recorrendo à metáfora do módulo de *hardware*, tipicamente o circuito integrado. É conceptualmente coincidente com o conceito de macrotransição atrás referido.

A especificação é constituída por diversas *páginas*, em que cada *página* armazena uma RdP Colorida. A sub-rede que se pretende modularizar é representada por uma transição em níveis hierárquicos superiores (isto é, nas *superpáginas*) e armazenada numa *página* autónoma da representação (denominada por *subpágina*). Os lugares presentes na *superpágina* interligados à *transição-a-substituir*<sup>24</sup> são denominados por *lugares-de-ligação* (“socket places”) enquanto que na *subpágina* se utilizam *lugares-porto* (“port places”) caracterizando o interface do módulo com o exterior.

---

<sup>24</sup> Embora a utilização do hífen nesta e noutras palavras semelhantes a utilizar ao longo deste trabalho possa ser sintacticamente discutível, foi considerada necessária para evidenciar o facto de se tratar da tradução de um termo técnico que de outra forma perderia expressividade.

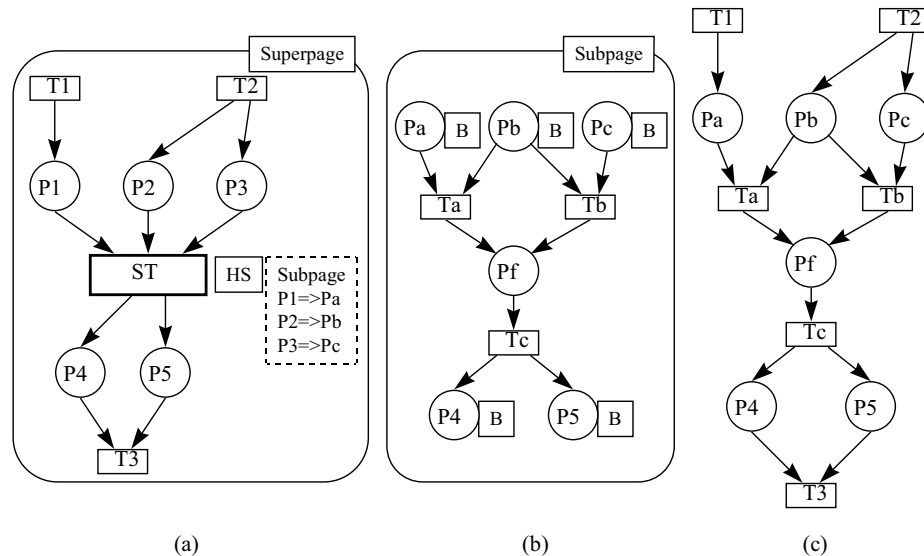


Figura 4-8 - Mecanismo de substituição de transição:

(a) superpágina; (b) subpágina; (c) modelo plano equivalente

Na Figura 4-8 apresenta-se um exemplo simples, ilustrando o mecanismo de substituição de transições. Na superpágina referida por *Superpage* (Figura 4-8(a)), utiliza-se uma referência a uma transição-a-substituir, ST, modelada na página referida por *Subpage* (Figura 4-8(b)). De notar que, associado à transição ST, se encontram várias anotações, nomeadamente HS para a identificar como uma transição Hierárquica de Substituição, bem como o nome da subpágina associada e a tabela de correspondência entre os lugares-de-ligação e os lugares-porto (caso o nome seja coincidente, pode ser omitido). Na subpágina os lugares-porto constituintes do interface são assinalados com a inscrição B (de “Border”); no sentido de suportar procedimentos de validação e apoio à especificação, existe uma outra característica (omitida na figura) que classifica o porto num dos seguintes tipos: entrada, saída, entrada/saída e geral (isto é, não especificado). O modelo plano resultante da fusão resultante do processo de integração do submodelo é apresentado na Figura 4-8(c).

De notar que o mecanismo simplesmente insere adequadamente o submodelo sempre que uma “transição hierárquica de substituição” é utilizada, “fundindo” os lugares-porto com os lugares-de-ligação. A subpágina pode ser utilizada tantas vezes quantas as desejadas, desde que se garanta ausência de referências recursivas.

Neste mecanismo não faz sentido considerar a transição-a-substituir (ST na Figura 4-8) como uma transição comum, para efeitos de análise, simulação ou execução da rede (dado que não se pode falar de transição habilitada ou de disparo da transição-a-substituir).

#### 4.2.2.a.2 Substituição de lugares

O segundo mecanismo proposto é o de substituição de lugares, que apresenta fortes semelhanças com o de substituição de transições. Neste caso, o módulo a ser fundido no modelo é representado por um lugar-a-substituir (que também não tem realidade do ponto de vista de execução do modelo), sendo referidas transições-de-ligação na superpágina que encontrarão as correspondentes transições-porto na subpágina que modela o módulo. A proposta apresentada prevê o tratamento automático de uma situação de subespecificação em que são utilizadas várias transições-de-ligação com uma referência igual a uma transição-porto; neste caso, procede-se à duplicação da transição-porto e arcos associados de forma a criar tantas transições-porto quantas as transições-de-ligação referidas.

Uma limitação significativa associada a esta forma de representar módulos (do ponto de vista de módulos de *hardware*) através de lugares-a-substituir e de transições-a-substituir é o facto de não ser permitida a sua interligação directa. Esta deverá ter sido a característica determinante para a não integração no modelo de RdP Coloridas Hierárquicas [Jensen, 92] do mecanismo de substituição-de-lugares. A limitação deve-se ao facto de considerar os nós-de-ligação apenas como nós virtuais, sem existência real do ponto de vista de execução do modelo (representam os nós-porto do módulo utilizado), conduzindo à impossibilidade de encontrar a rede colorida não-hierárquica correspondente à interligação de dois “macronós”.

A associação dos dois mecanismos apresentados a mecanismos disponíveis nas linguagens de programação de sistemas é imediata, nomeadamente às técnicas de utilização de macros em linguagem *assembly* ou de procedimentos *in-line* nalgumas linguagens de alto-nível. A busca de mecanismos semelhantes à utilização de subprogramas, aparece, assim, como óbvia.

#### 4.2.2.a.3 Evocação de transições

O terceiro mecanismo proposto endereça precisamente a funcionalidade apresentada e é concretizado através da evocação de transições. A notação gráfica utilizada para representar a transição-a-evocar é semelhante à utilizada para a transição-a-substituir; no entanto, neste caso faz-se referência ao atributo HI, de hierárquico (“Hierarchical”) e evocação (“Invocation”), em vez de HS. A subpágina associada tem as mesmas características que os módulos representados por uma transição-a-substituir, às quais se adicionou uma nova característica identificadora da terminação da execução do “subprograma”; pode ser associada a lugares e transições e concretiza-se na inclusão da anotação “*Exit*” junto ao nó respectivo.

A semântica associada pode ser sintetizada da seguinte forma: quando a transição-a-evocar está habilitada e ocorre, as marcas presentes nos lugares de entrada da transição-a-evocar são transferidas para os lugares-porto de entrada da subpágina associada; a evolução dos modelos associados à superpágina e à subpágina decorre concorrentemente até que alguma transição da subpágina com o atributo “Exit” dispare ou algum lugar da subpágina com o atributo “Exit” receba uma marca; em qualquer destas situações, as marcas presentes nos lugares-porto de saída (com o atributo “Out”) são transferidas para os lugares de saída correspondentes da transição-a-evocar na superpágina e cessa a execução do modelo representado na subpágina.

Os lugares interligados à transição-a-evocar contidos na superpágina armazenam os parâmetros actuais da sub-rotina, representada pela transição-a-evocar, no sentido de chamada por valor, utilizada em linguagens de alto-nível; desta forma, os lugares ligados à transição-a-evocar são designados por *lugares-com-parâmetros*.

São permitidas referências circulares em páginas contendo transições-a-evocar, isto é, a recursividade, neste caso, é permitida.

#### **4.2.2.a.4 Fusão de lugares e fusão de transições**

Finalmente, os dois últimos mecanismos propostos baseiam-se na fusão de conjuntos de nós do mesmo tipo, podendo, assim, ocorrer a fusão de lugares e a fusão de transições. Como referido, a fusão de lugares é parte integrante da definição das RdP Coloridas Hierárquicas.

A idéia subjacente é a de permitir ao projectista justapor um conjunto de nós, representados separadamente numa ou em várias páginas, considerando-os como um único nó. No caso da fusão de lugares, corresponde a, do ponto de vista de execução do modelo, ligar rigidamente a marcação de cada um dos lugares pertencentes ao conjunto, de modo que alterações ocorridas num dos lugares são implicitamente reflectidas nos restantes lugares do conjunto.

A fusão de transições permite sincronizar o disparo de um conjunto de transições.

Quando aplicada a nós contidos numa mesma página com uma única instância, o mecanismo de fusão é, fundamentalmente, um expediente que permite uma simplificação da representação gráfica, evitando, por exemplo, arcos muito longos ou de difícil apreensão.

#### 4.2.2.a.5 Canais de comunicação síncrona

Um mecanismo complementar de estruturação modular de RdP Coloridas (e RdP Coloridas Hierárquicas) foi proposto em [Christensen & Hansen, 94]. Baseia-se no conceito de canais de comunicação síncrona em que podem estar envolvidas duas transições. Na Figura 4-9 apresenta-se um exemplo elementar, extraído de [Christensen & Hansen, 94], ilustrando a decomposição de uma RdP Colorida em dois módulos que comunicam através de um canal síncrono e que são comportamentalmente equivalentes à RdP inicial.

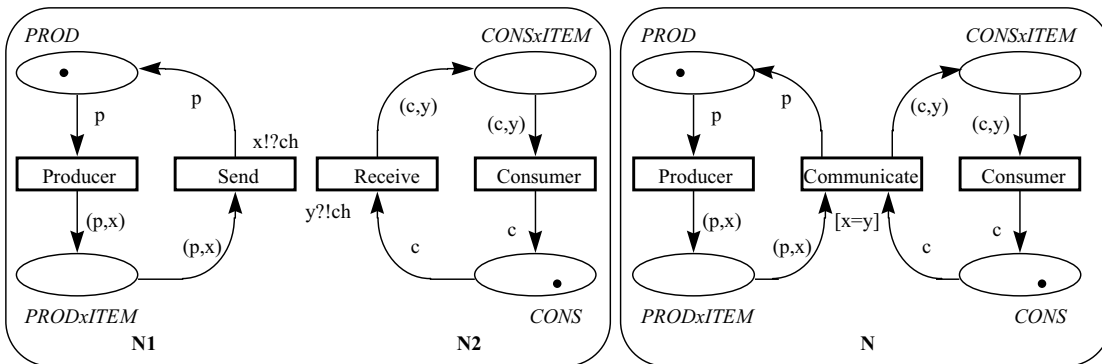


Figura 4-9 - Utilização de canais síncronos e RdP equivalente.

Embora no exemplo da Figura 4-9, não fique clara a diferença entre a utilização de transições com canais síncronos e do mecanismo de fusão de transições, os dois conceitos diferem substancialmente. No caso dos canais síncronos, a comunicação é estabelecida, de modo bi-direccional, entre duas transições, satisfazendo os vínculos (“bindings”) a elas associados; as transições envolvidas na comunicação, para além de utilizarem o mesmo canal síncrono, dispõem de um atributo referido graficamente por  $?!$  e  $!?$ . Este atributo permite dividir o conjunto das transições que se referem a um determinado canal, em dois conjuntos disjuntos, de onde se elegem as duas transições a envolver na comunicação. Do que ficou dito se conclui que, existindo  $n+m$  transições referindo o canal  $ch$  (em que  $n$  transições dispõem do atributo  $!?$  e  $m$  transições do atributo  $?!$ ), a RdP Colorida comportamentalmente equivalente seria obtida utilizando  $n*m$  transições em substituição das  $n+m$  transições com canais síncronos. Quando  $n$  e  $m$  forem pequenos, os benefícios desta técnica são discutíveis; quando  $n$  e  $m$  forem elevados, obter-se-á uma redução de transições, que aliada à atitude modular, permitirá compactar a representação, não sendo, no entanto, claro que a legibilidade intuitiva do modelo resultante fique beneficiada.

#### 4.2.2.b Utilizando RdP Predicado-Transição

Em [He, 96] são formalmente definidas as RdP Predicado-Transição Hierárquicas. Neste trabalho, a relação hierárquica entre os vários módulos utilizados é garantida por construção, sendo as entidades de nível superior designadas por superlugares e supertransições. A sua interligação directa é permitida. Três características são, do ponto de vista do trabalho que aqui se apresenta, interessantes de referir:

- a primeira prende-se com a notação gráfica utilizada para representar os superlugares e supertransições: uma linha tracejada substitui a linha a cheio quer nas elipses representativas dos lugares, quer nos rectângulos representativos das transições; a utilização de uma elipse a tracejado é realizada quer para representar o superlugar no modelo superior, quer para envolver a sub-rede dependente; de modo semelhante, a utilização de um rectângulo a tracejado é realizada quer para representar a supertransição no modelo superior, quer para envolver a sub-rede dependente;
- na representação de sub-redes são utilizados arcos responsáveis pelo interface que se ligam directamente ao contorno da sub-rede, isto é, arcos que se encontram “abertos” numa das suas extremidades, sendo “ligados” apenas quando se considera a inserção da sub-rede no modelo onde o supernó associado está referido; os “statecharts” são referidos como motivação para esta atitude;
- a terceira característica, porventura a mais interessante e sui generis, utiliza a inscrição associada a um arco para permitir a representação compactada de um conjunto de arcos; a inscrição define não só o tipo e número de marcas “transportadas” pelo arco, como também identifica os arcos componentes (subarcos) a que cada marca está associada; a identificação dos arcos componentes é adicionada à referência da marca (como, por exemplo, em  $\langle l_1, x \rangle$  para representar a inscrição  $x$  associada ao arco componente  $l_1$ ). Na inscrição do arco composto são utilizados dois construtores, representados por + (“ou”) e \* (“e”) e expressando não-determinismo e simultaneidade, respectivamente.

A Figura 4-10 apresenta um exemplo simples, utilizado em [He, 96], em que as características referidas são ilustradas. A inscrição no arco de  $t_1$  para  $p_1$  ( $\langle l_1, x \rangle * \langle l_2, y \rangle$ ) revela que este arco representa um conjunto de dois arcos, com identificadores  $l_1$  e  $l_2$  respectivamente, aos quais se afectam as inscrições  $x$  e  $y$ , em que a “circulação” de marcas é concorrente, isto é, simultânea; a sua tradução para RdP PrT apresentada na Figura 4-10(b) dá origem aos arcos de  $t_1$  para  $p_3$  e de  $t_1$  para  $p_4$ .

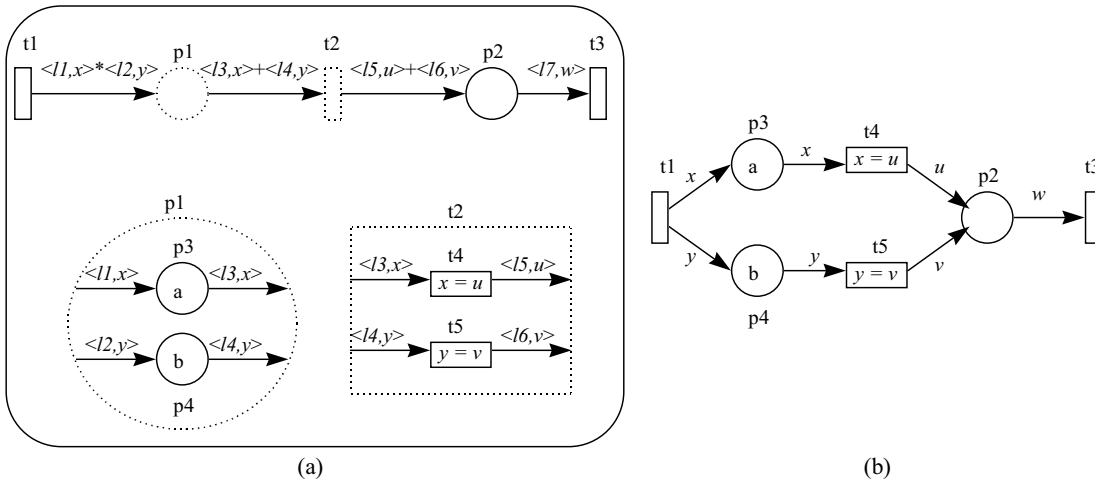


Figura 4-10 - Uma RdP Predicado Transição Hierárquica (a) e Rdp PrT equivalente (b).

Os trabalhos que estiveram na base da referida proposta foram, fundamentalmente, [Reisig, 87] e [He & Lee, 91]. Em [He & Lee, 91] é proposta uma metodologia de construção de especificações com RdP Predicado-Transição, em que a notação apresentada é introduzida e onde se utilizam quatro técnicas de transformação, nomeadamente refinamento, abstracção, síntese e decomposição. A metodologia introduz hierarquia e modularidade nas especificações produzidas. As técnicas de refinamento e abstracção suportam os paradigmas associados às abordagens de especificação hierárquica descendente e ascendente, respectivamente. As técnicas de síntese e decomposição permitem que, num mesmo nível de abstracção, se realize a composição de uma RdP a partir da interligação de várias RdP (síntese) ou a sua desagregação em várias RdP ou módulos (decomposição). Deste modo suportam-se os paradigmas associados à programação modular e técnicas de “dividir para reinar”.

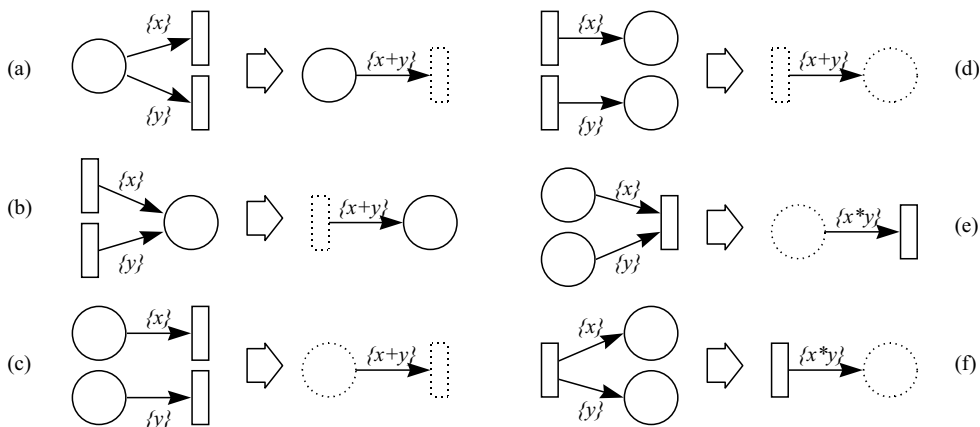


Figura 4-11 - Os seis padrões básicos de suporte à abstracção.

As inscrições utilizadas durante operações de abstracção (hierárquica ascendente) associadas aos arcos que representam arcos múltiplos são caracterizadas em seis estruturas básicas que se apresentam

na Figura 4-11. Nesta figura torna-se explícita a semântica associada aos símbolos + e \* utilizados nas inscrições dos arcos.

O operador \* tem precedência sobre o operador +, permitindo a utilização iterativa coerente de técnicas de abstracção sobre um modelo. A interligação entre RdP em processos envolvendo síntese e decomposição utiliza inscrições semelhantes. Contudo a notação gráfica utilizada não mantém, para todos os casos possíveis, uma relação bidireccional unívoca entre as representações resultantes de acções de síntese e abstracção (ver [He, 96]); como consequência, a leitura de páginas associadas a níveis de abstracção mais elevados podem não explicitar relações de dependência existentes entre arcos, nomeadamente as relações expressas pelos operadores + e \*.

#### *4.2.2.c Utilizando representações orientadas por objectos*

Um dos paradigmas de suporte ao desenvolvimento de novas metodologias de análise e ambientes de desenvolvimento associados a sistemas de elevada complexidade é, sem dúvida, o da orientação por objectos. As características intrínsecas de modelação de mecanismos de herança, encapsulamento e polimorfismo têm permitido suportar a especificação, validação e teste modulares, melhorar os índices de produtividade devido às características de reutilização disponíveis, bem como os níveis de qualidade dos produtos produzidos.

Vários trabalhos têm vindo a integrar os conceitos da orientação por objectos nos formalismos das RdP. Diversas têm sido as abordagens propostas, bem como os objectivos perseguidos. Nalguns casos pretende-se utilizar o paradigma da orientação por objectos para reforçar as capacidades de modelação das RdP, noutros a abordagem é dual, pretendendo-se utilizar as RdP para modelar ou implementar linguagens concorrentes ou metodologias orientadas por objectos. O segundo grupo conduz, normalmente, à impossibilidade de aplicar métodos de análise comuns nas RdP de baixo-nível. Em qualquer dos grupos, contudo, enfatiza-se a componente “inscrições na rede” para embeber estruturas de dados e processamentos associados, em detrimento da representação gráfica explícita.

Nos parágrafos seguintes as duas linhas de trabalhos identificadas são, brevemente, apresentadas.

Os trabalhos de Charles Lakos, integrando-se no primeiro grupo, apresentam-se como o caso mais suportado, do ponto de vista da formulação teórica, nesta área. Como defendido em [Lakos, 96], a resposta à modelação de sistemas cada vez mais complexos encontra-se na progressão natural que se

inicia nas RdP Lugar-Transição e é continuada nas RdP Coloridas e nas RdP Coloridas Hierárquicas, sendo as RdP Objecto<sup>25</sup> (abreviadamente OPIIN) o estado actual dessa evolução.

A transição das RdP Coloridas para as RdP Objecto é apresentada em [Lakos, 95]; um conjunto de classes intermédias é utilizado para justificar e “suavizar” a transição, sendo provada a equivalência comportamental das novas classes em relação às definidas anteriormente. A primeira expansão a partir das RdP Coloridas conduz às RdP Coloridas Modulares (MCPN), onde se suporta a utilização de supertransições e superlugares (conceito complementar à substituição-de-transições das RdP Coloridas Hierárquicas), as RdP Baseadas em Objectos (OBPN), onde se suportam vários níveis de actividade e as marcas deixam de ser objectos passivos podendo encapsular sub-redes com actividade própria, as RdP Orientadas por Objectos (OOPN), onde se adicionam as noções de herança, polimorfismo e vínculo (“*binding*”) dinâmico, chegando finalmente às RdP Object (OPN), resultantes da inclusão de arcos de teste e inibidor. Este modelo foi posteriormente reformulado, em [Lakos, 96], passando a receber a abreviatura OPIIN e onde a utilização de herança e polimorfismo é “implementada de modo mais elegante” (sic). Como linguagens de suporte ao desenvolvimento dos referidos modelos foram utilizadas duas linguagens: LOOPN para as OOPN e LOOPN++ para as OPN [Lakos & Keen, 94].

De particular relevância para o presente trabalho, refira-se a utilização de superlugares, concomitantemente com a utilização de supertransições (semelhantes à substituição-de-transições das RdP Coloridas Hierárquicas). A utilização de canais síncronos, propostos em [Christensen & Hansen, 94], implementa a interacção com os superlugares. A Figura 4-12, retirada de [Lakos, 95], ilustra o processo.

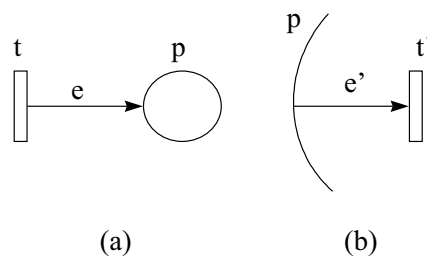


Figura 4-12 - Vistas externa e interna de um superlugar.

Do ponto de vista externo (superpágina), o multiconjunto  $e$  é oferecido a  $p$ , através do arco de  $t$  a  $p$ . Do ponto de vista interno ao superlugar, o multiconjunto  $e$  é recebido através do disparo da transição  $t'$  interna; a união das marcas oferecidas, representadas por  $e$ , com as marcas recebidas, representadas por  $e'$ , é realizada através de canais síncronos.

<sup>25</sup> O termo “RdP Objecto” é adoptado como tradução de “Object Petri nets”.

Nos parágrafos seguintes, apresentam-se, de forma necessariamente incompleta, algumas referências para a segunda linha de trabalhos referida, utilizando as RdP embebidas em metodologias ou linguagens de programação orientadas por objectos. Alguns dos trabalhos a referir foram desenvolvidos em áreas de aplicação específicas e possuem protótipos disponíveis publicamente. Considerando que a ênfase do presente trabalho não se coaduna com uma classificação neste segundo grupo de trabalhos, apenas se faz uma breve referência a cada um dos cinco trabalhos considerados mais significativos:

- i) a linguagem CLOWN (CLass Orientation With Nets) [Battiston & Cindio, 93], que, sendo parte integrada no ambiente de desenvolvimento ONE, utiliza o modelo das RdP OBJSA [Battiston, Cindio & Mauri, 88]; a execução do modelo baseia-se na decomposição em componentes elementares que são máquina de estados (de onde herda “SA” no nome, de “*Superposed Automata*”, apresentado anteriormente neste capítulo) e em que as marcas são definidas como tipos de dados abstractos utilizando a linguagem OBJ2 (de onde herda “OBJ” no nome);
- ii) o formalismo denominado CO-OPN (Concurrent Object Oriented Petri net) [Buchs & Guelfi, 91] em que se abordam com especial ênfase as questões de subtipos e de subclasses;
- iii) tendo também a sua génese em áreas de aplicação onde o paradigma cliente-servidor se utiliza, em [Sibertin-Blanc, 93], apresenta-se uma proposta de caracterização modular dos modelos associados, em que a comunicação entre servidores e clientes é assíncrona, também implementada através de lugares modelando canais uni-direccionais. Posteriormente, em [Sibertin-Blanc, 94], as RdP Cooperativas (Cooperative Nets) são propostas. A linguagem SYROCO [Sibertin-Blanc, Hameurlain & Touzeau, 95], implementada em C++, integra esses trabalhos com os COO (CoOperative Objects) [Bastide & Palanque, 93], também de investigadores de Toulouse. Alguns dos resultados obtidos nestes grupos têm vindo a aplicar as RdP ao desenho de interfaces de utilizador controlados por eventos (“event-driven”) [Bastide & Palanque, 95];
- iv) outro conjunto de trabalhos com contribuições significativas nas três áreas identificadas por Kurt Jensen como “chave do sucesso” (teoria, ferramentas e aplicações) usa como linguagem de especificação uma classe de RdP de alto nível denominada THORN (Timed Hierarchical Object-Related Nets) [Schöf, Sonnenschein & Wieting, 95]. O conjunto de ferramentas desenvolvido utiliza esta classe de RdP como linguagem de simulação, dado que, como referido em [Schöf, Sonnenschein & Wieting, 95], devido à complexidade dos conceitos utilizados a análise formal dos modelos THORN é praticamente impossível. Os

conceitos da programação orientada por objectos são utilizados extensivamente, sendo a linguagem C++ utilizada para a implementação. Os lugares podem ser estruturados como multiconjuntos, pilhas, filas ou filas com prioridades; contêm marcas de apenas um tipo. As marcas são objectos descritos em C++. As transições têm associado uma condição de activação, um bloco de acção e duas funções temporais (definidas em C++) através das quais se especificam os tempo de disparo e tempo de atraso associados à transição. Finalmente, os arcos podem ser de vários tipos; para além de “comuns”, podem ser de habilitação (na perspectiva dos arcos de teste doutros trabalhos, [Lakos & Christensen, 94] por exemplo), de inibição e de consumo (caso em que são utilizadas todas as marcas presentes no lugar de origem do arco). Do ponto de vista de estruturação do modelo, dois mecanismos são considerados, ambos associados às transições: refinamento de transições e evocação de sub-redes. A semântica utilizada é semelhante à utilizada em [Huber, Jensen & Shapiro, 90] para os seus mecanismos de substituição de transições e evocação de transições, pelo que não serão objecto de comentários adicionais;

- v) o ambiente de desenvolvimento de aplicações para arquitecturas concorrentes denominado CAB (Concurrent Applications Builder) [Bruno et al., 95], suportado pela linguagem Protob [Baldassari & Bruno, 88], integra técnicas comuns nas abordagens de análise de sistemas orientadas por objectos. A própria linguagem Protob é baseada em redes temporizadas de alto-nível orientadas por objectos permitindo a representação de concorrência e sincronização. Os domínios de aplicação endereçados pela linguagem incluem os sistemas embebidos de tempo-real, os sistemas de telecomunicações e sistemas de manufactura; na realidade, domínios de aplicação próximos ou mesmo coincidentes com o do presente trabalho. Uma aplicação escrita em Protob é uma colecção de objectos comunicantes, implementáveis através de instâncias de classes. Associada a uma instância de classe está a sua parte gráfica, isto é a RdP. Aos lugares estão associados um identificador e um determinado tipo de marca, podendo conter qualquer número de marcas desse tipo, armazenadas através de filas com base no seu tempo de chegada (e não de conjuntos ou multiconjuntos, como normalmente). Não são permitidos pesos associados aos arcos e podem ser associadas restrições de tempo às transições, respeitando uma de duas disciplinas: disparo atrasado<sup>26</sup> e libertação atrasada<sup>27</sup>.

---

<sup>26</sup> “Delayed firing” em [Bruno et al., 95].

<sup>27</sup> “Delayed release” em [Bruno et al., 95].

### 4.2.3 Utilizando RdP não-autónomas

Considerando o conjunto de classes de RdP que integram características não-autónomas, se excluídas forem as associadas às propostas de RdP Estocásticas de reduzida relevância para o presente trabalho, não são conhecidas propostas específicas aplicáveis para a gestão da complexidade **específica** de modelos de RdP não-autónomas, para além de [Gomes & Steiger-Garção, 96c] e que será embebida nas propostas a apresentar posteriormente neste capítulo.

Em [Gomes & Steiger-Garção, 96c] utilizam-se extensivamente os mecanismos de representação hierárquica utilizados nos statecharts [Harel, 87] para a representação de máquinas de estado hierarquicamente organizadas.

São, no entanto, vários os trabalhos integrando RdP não-autónomas e suportando hierarquia. Recorrem, normalmente, a macronós e baseiam-se em classes específicas de RdP Ordinárias ou Generalizadas, logo com capacidades de modelação limitadas; estão associados a áreas de aplicação típicas como controlo de sistemas de manufactura e de sistemas digitais. Como exemplos refiram-se [Silva & Valette, 89] e [Machado, 96], que utiliza o trabalho inicialmente proposto em [Fernandes, 94].

Com especial interesse para este trabalho, refira-se a Tese de Mestrado [Barros, 96] que, partindo do modelo proposto em [Gomes & Steiger-Garção, 92] e [Gomes & Steiger-Garção, 95d] e da experiência acumulada em [Luis & Ventura, 92] e em [Barros, Silva & Soares, 93], propõe uma classe de RdP Coloridas Não-Autónomas, denominada CpPNeTS. As CpPNeTS suportam a modelação de macrotransições e macrolugares, bem como a sua interligação directa, cuja representação é realizada em C++ e em relação à qual é disponibilizada uma biblioteca para a sua execução e análise através do espaço de estados associado.

### 4.2.4 Ênfases e lacunas dos trabalhos apresentados

Nos parágrafos seguintes tenta-se proceder a um breve balanço-resumo das características principais das várias soluções apresentadas para a gestão da complexidade de modelos RdP, no sentido da identificação de algumas questões deixadas em aberto, em que as propostas são inexistentes, incompletas ou ambíguas. As RdP Reactivas Hierárquicas (abreviadamente referidas por RdP-RH) pretendem resolver este conjunto de questões.

Como primeira característica notável neste balanço, refira-se o reduzidíssimo número de propostas específicas para a gestão de RdP Não-Autónomas. As técnicas comuns de estruturação do modelo de RdP (com excepção de [Gomes & Steiger-Garção, 96c]) consideram unicamente a componente

autónoma do modelo. Na área de especificação de controlo para sistemas de tempo-real, embora as RdP Interpretadas sejam utilizadas como uma referência, não se conhecem outros trabalhos que abordem as questões específicas associadas ao tratamento dos eventos externos e sua sincronização num contexto de estruturação hierárquica ou modular do sistema. Como exemplos de questões em aberto nesta área, refiram-se o suporte para a modelação de interrupções ou de atribuição de prioridades a vários processos concorrentes.

É uma **primeira lacuna** identificada à qual as RdP-RH pretendem dar resposta.

Do ponto de vista do suporte à especificação hierárquica e modular, os mecanismos de suporte ao refinamento/abstracção associados às abordagens descendente e ascendente, são normalmente implementados através de macronós (ou supernós, como designados nalguns trabalhos), nomeadamente macrolugares e macrotransições, representando sub-redes em que a interligação ao exterior é garantida através de nós de um único tipo (lugares ou transições). A utilização de módulos com uma caracterização externa composta por lugares e transições encontra-se num número reduzido de propostas. A interligação directa entre macronós nem sempre é permitida, muito embora se apresente como necessária para o suporte às técnicas de síntese (ou agregação) e decomposição.

A execução, simulação ou análise dos modelos utilizando macronós necessitam de considerar o modelo plano, isto é, o modelo resultante da substituição dos macronós pelas sub-redes representadas. Deste modo, a utilização de técnicas de estruturação do modelo são efectivas apenas ao nível da especificação gráfica do modelo, importando, pois, encontrar mecanismos que permitam a análise, simulação ou execução do modelo com base nos seus submodelos constituintes, considerando vários níveis de abstracção e dando suporte efectivo a uma modelação multinível.

É uma **segunda lacuna** identificada à qual as RdP-RH pretendem dar resposta.

Um outro aspecto que necessita de esclarecimento complementar relativo à semântica associada à representação gráfica de macronós, colocado em [He, 96], é ilustrado na Figura 4-13.

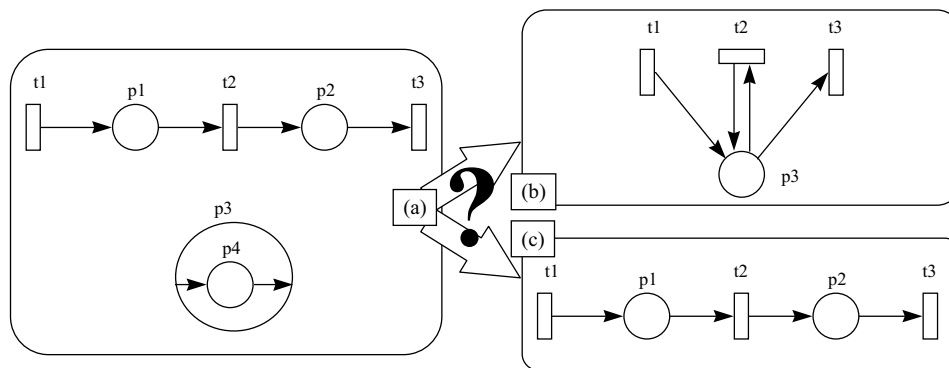


Figura 4-13 - Notação gráfica ambígua: a) especificação;  
b) interpretação como fusão de lugares; c) interpretação como substituição de lugares.

Na Figura 4-13(a), os lugares p1 e p2 são implementados através da página referida por p3, contendo neste caso elementar unicamente o lugar p4. Se a semântica associada for coincidente com a fusão de lugares, então o modelo resultante é o apresentada na Figura 4-13(b); por outro lado, se a semântica for a associada à substituição de lugares, então o modelo associado é o apresentado na Figura 4-13(c). Em [Huber, Jensen & Shapiro, 90] esta ambiguidade é resolvida através da utilização de atributos de texto inseridos na representação gráfica e associados aos lugares envolvidos. Caso a inscrição seja FP (“fusion of places”) então o que se pretende modelar é a situação da Figura 4-13(b); caso se pretenda modelar a situação da Figura 4-13(c), então a inscrição deverá ser HS (“hierarchical substitution”).

A implementação de algumas técnicas (nomeadamente nas abordagens orientadas por objectos e na evocação de transição proposto em [Huber, Jensen & Shapiro, 90]), não raras vezes necessita da criação e destruição dinâmica de sub-RdP. No entanto, a alocação dinâmica de memória necessária para a sua implementação é uma característica não desejada do ponto de vista da implementação de sistemas de tempo-real, em que o determinismo na execução e a quantificação a priori dos recursos necessários à execução são requisitos a satisfazer.

Este é o **terceiro aspecto** identificado ao qual as RdP-RH pretendem dar resposta.

Do ponto de vista da interligação entre os vários módulos (ou sub-redes) constituintes do modelo global do sistema, podem ser identificadas várias soluções baseadas em comunicações síncronas ou em comunicações assíncronas. No primeiro grupo encontram-se as soluções baseadas em máquinas de estados cuja evolução é sincronizada através de transições comuns, como o modelo de “Superposed Automata”, SA, utilizado em [Battiston & Cindio, 93], ou na utilização de canais síncronos, como proposto em [Christensen & Hansen, 94] e utilizado nas RdP Objecto [Lakos, 95]. No segundo grupo encontram-se soluções em que a comunicação é garantida através de lugares comuns, como em [Bruno et al., 95]. A solução proposta nas RdP-RH enquadra-se no primeiro grupo de soluções.

Como observação final, uma chamada de atenção para os trabalhos que apresentam uma notação gráfica enriquecida, no sentido de exibirem explicitamente características comportamentais ou de interligação entre módulos que se pretendem modelar; como trabalhos considerados mais significativos refiram-se [Lakos & Christensen, 94] e [He & Lee, 91]. Também para o caso das RdP Não-Autónomas, onde essas características são particularmente significativas, não se conhecem notações gráficas especificamente aplicáveis.

Esta é a **quarta lacuna** identificada à qual as RdP-R pretendem dar resposta.

Desta forma, fica claro que, muito embora muito trabalho significativo tenha sido já realizado em sintonia com avanços verificados noutras áreas das Ciências de Engenharia, ainda existe um amplo campo onde as soluções encontradas para gerir a complexidade de um modelo são insuficientes ou inadequadas a áreas de aplicação específica, como a dos sistemas reactivos de tempo-real, e necessitam de ser melhoradas.

A próxima secção apresenta trabalho original nessa área.

### ***4.3 Propostas para a modelação hierárquica e modular de sistemas reactivos de tempo-real***

As técnicas de estruturação do modelo que se irão propôr seguidamente, perseguem vários dos objectivos apresentados na secção anterior, nomeadamente a obtenção de suporte efectivo:

- à especificação modular e hierárquica;
- à implementação modular e incremental,

através de modelos:

- mais compactos;
- mais expressivos (no sentido de facilmente legíveis);
- que permitam a obtenção de mais e melhor documentação de projecto.

As técnicas a propôr estão, conforme referido, claramente vocacionadas para a modelação de sistemas reactivos de tempo-real, onde se exige um comportamento determinístico bem como a caracterização a priori do conjunto de recursos necessários associados à sua execução.

Propõe-se a utilização de três técnicas de decomposição aplicáveis aos nós do grafo, a saber:

- decomposição horizontal,
- decomposição vertical e
- decomposição mista.

Como característica comum às diferentes técnicas propostas, refira-se o conceito de página, como objecto representando uma RdP. As páginas que contêm referências a outras páginas (ou seja, a outras RdP), isto é, que ocupam uma posição mais elevada na hierarquia, são designadas por superpáginas, enquanto que as páginas referidas (hierarquicamente dependentes das primeiras) são denominadas por subpáginas. Esta classificação é coincidente com a terminologia utilizada nas RdP Coloridas Hierárquicas.

Os três tipos de técnicas de decomposição utilizam diferentes tipos de nós. Para permitir sempre que necessário uma referência não ambigua, mesmo antes da sua apresentação detalhada nas secções seguintes, introduz-se a seguinte terminologia:

- os nós especiais utilizados na decomposição horizontal são denominados macronós; podem ser macrolugares, macrotransições ou macroblocos; as subpáginas associadas aos macronós designam-se por macro-redes;
- os nós especiais utilizados na decomposição vertical são denominados metanós; podem ser metalugares ou metatransições; as subpáginas associadas aos metanós designam-se por meta-redes;
- os nós especiais utilizados na decomposição mista são denominados supernós; podem ser superlugares ou supertransições; as subpáginas associadas aos supernós designam-se por super-redes;

Nas subsecções seguintes são caracterizados cada um dos mecanismos propostos.

#### **4.3.1 Decomposição horizontal**

O mecanismo de decomposição horizontal é definido no sentido “normalmente” utilizado para suportar abordagens modulares, nomeadamente as abordagens descendentes e ascendentes, isto é, através de refinamentos (ou agregações) sucessivos(as). Baseia-se na utilização de macronós.

Cada nível da representação é associado a uma página; essa página pode conter referências a subpáginas; cada subpágina, associada a um macronó, é, aqui, denominada por macro-rede e está, normalmente, contida numa página autónoma da representação gráfica (isto é, numa folha).

Neste tipo de decomposição, para que a execução ou análise do modelo se realize, torna-se necessário obter previamente o modelo plano associado, isto é, aquele em que todas as referências a macronós se encontram substituídas pelas páginas associadas às respectivas macro-redes. A execução do modelo é realizada com base numa representação com um único nível, horizontal. Esta é a razão da denominação proposta para este tipo de técnica, de decomposição horizontal.

O interface das macro-redes com o exterior, isto é, a indicação do modo como a macro-rede se interliga no modelo onde é referido, é efectuada através de portos ou nós-fronteira. Do ponto de vista da execução do modelo, estes portos são apenas utilizados como referências que permitem a inserção adequada da macro-rede no modelo ou modelos em que possa estar referida. Mais à frente este assunto será retomado.

São consideradas três possibilidades para proceder a uma decomposição de tipo horizontal:

- utilizando macrolugares;
- utilizando macrotransições;
- utilizando macroblocos.

No que se segue, quer macrolugares, macrotransições ou macroblocos podem ser referidos por macronós.

Os macrolugares representam subpáginas em que o interface é constituído por nós de um único tipo, concretamente lugares. São, portanto, macro-redes onde chegam e partem arcos interligados a transições, macrotransições ou portos-transição de macroblocos.

As macrotransições representam subpáginas em que o interface é constituído apenas por transições. São macro-redes onde chegam e partem arcos interligados a lugares, macrolugares ou portos-lugar de macroblocos.

A introdução do conceito de macrobloco pretende suportar integralmente a atitude de decomposição modular de um sistema, em situações que cada módulo não pode ser caracterizado tão simplesmente como um macrolugar ou uma macrotransição, isto é, em situações que o interface necessita de ser constituído simultaneamente por lugares-fronteira e por transições-fronteira.

Poderia ser considerado meramente como um expediente gráfico que destrói a característica fundamental de grafo bipartido associada às RdP. No entanto, é mais importante e menos grave do que isso: corresponde a um instrumento que é disponibilizado ao projectista para a decomposição de um modelo, particularmente aos níveis de abstracção mais elevada. Como a análise e execução das RdP-R se verifica apenas no modelo plano, também a característica de grafo bipartido é mantida. A sua utilização não adiciona compromissos à interpretação intuitiva do modelo, para além dos introduzidos pelos macrolugares e macrotransições.

Deste modo, os macrolugares e as macrotransições podem ser considerados como casos particulares de macroblocos.

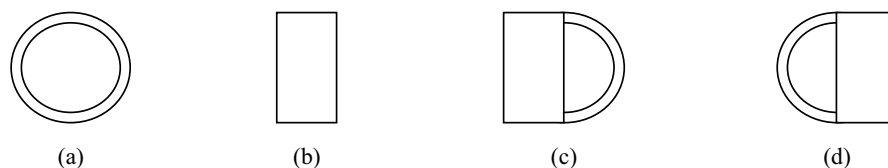


Figura 4-14 - Notação gráfica de macronós:

a) macrolugar, b) macrotransição, c) e d) macrobloco

Do ponto de vista da representação gráfica, utilizam-se as seguintes convenções (ver Figura 4-14):

- um macrolugar é representado como um lugar mas utilizando uma linha dupla;
- uma macrotransição é representado através de um rectângulo;
- um macrobloco é representado através de uma composição, metade reservada a receber arcos provenientes de transições ou portos-transição, representada como um macrolugar e outra metade reservada a receber arcos provenientes de lugares ou portos-lugar, representada como uma macrotransição.

Os mecanismos utilizados, que se descrevem nas subsecções seguintes, permitem a substituição dos macronós pelas macro-redes associadas, numa atitude semelhante à utilizada pelas técnicas de módulos quando se desenvolve *hardware* ou de “macros” quando se utiliza linguagem “assembly”, isto é, o submodelo será replicado localmente e na sua totalidade em cada evocação realizada. Podem ser utilizadas múltiplas instâncias de um macronó, desde que se garanta que o modelo resultante não contem referências circulares envolvendo um mesmo nó (a recursividade não é, obviamente, suportada).

Nas subsecções seguintes são apresentados alguns exemplos e definições com o objectivo de esclarecer sobre as semânticas associadas a cada uma das estruturas propostas de macronó.

### 4.3.1.a Utilizando macrolugares

Como referido, o macrolugar corresponde a um macronó em que todos os nós da fronteira são do tipo lugar. A Figura 4-15 apresenta um exemplo básico em que se pretende modelar a utilização de uma sequência de controlo através de um macrolugar; as inscrições associadas a arcos e transições, devidas à utilização de características coloridas e não-autónomas, embora não estejam explicitamente representadas, não sofrem alterações devidas ao mecanismo de decomposição/agregação.

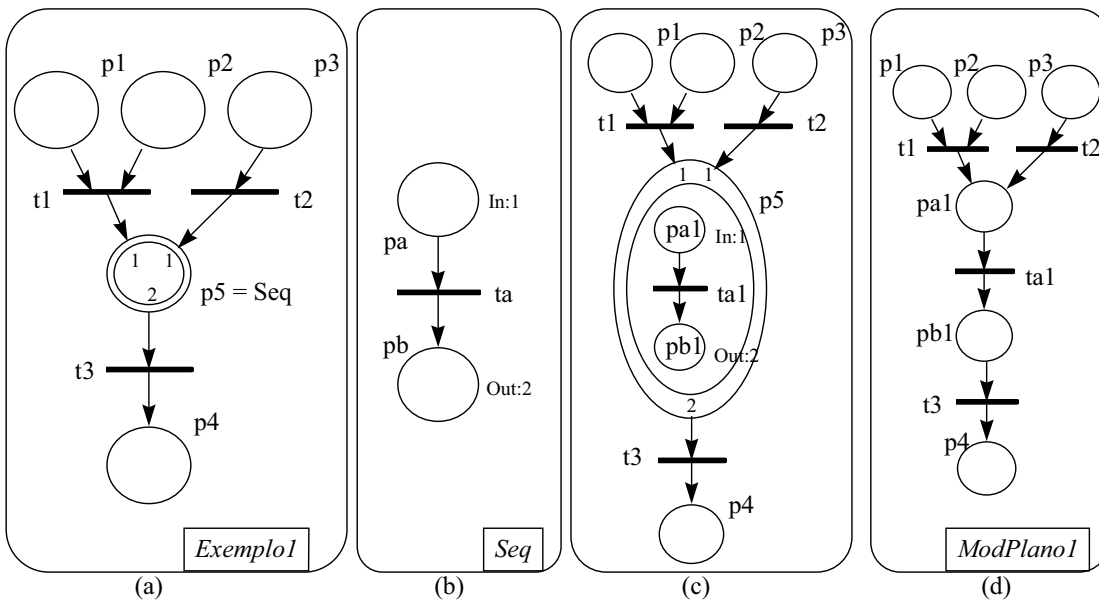


Figura 4-15 - Modelação de uma sequência de controlo através de um macrolugar:

a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente.

Na Figura 4-15(a) apresenta-se a página de topo da representação, incluindo uma referência ao macrolugar p5, implementado através da subpágina *Seq*. As referências aos lugares-fronteira utilizados (“1” e “2” no caso apresentado) são realizadas de forma explícita.

Dois métodos básicos podem ser utilizados para indicar as interligações entre a superpágina e a subpágina (dependendo do editor gráfico a utilizar):

- na subpágina, identificação dos lugares constituintes do interface, através de um atributo especial (que pode tomar os valores “In”, “Out” ou “In/Out”) e de um número de ordem associado; na superpágina, referência junto a cada arco dos números de ordem associados ao porto de entrada na subpágina; este método está próximo do normalmente utilizado em editores de sistemas electrónicos e foi utilizado em protótipos académicos de editores desenvolvidos em projectos de fim de curso orientados pelo autor do presente trabalho [Luís & Ventura, 92] [Barros, Silva & Soares, 93];

- na subpágina, identificação dos lugares constituintes do interface, através de um atributo especial (que pode tomar os valores “In”, “Out” ou “In/Out”); na superpágina, é introduzida uma tabela de referências cruzadas, incluindo todos os arcos ligados ao macrolugar com os identificadores utilizados na superpágina e subpágina; no exemplo apresentado seria utilizada a seguinte lista:  $t1 \rightarrow pa$ ;  $t2 \rightarrow pa$ ;  $pb \rightarrow t3$ ; este método está próximo do utilizado no ambiente Design/CPN para as macrotransições (o conceito de macrolugar não é suportado nesse ambiente).

No exemplo apresentado na Figura 4-15 utilizou-se o primeiro dos métodos referidos.

Como explicitamente apresentado na Figura 4-15, é possível utilizar na superpágina arcos múltiplos associados a um único porto de entrada ou saída. Este assunto será retomado em subsecção posterior.

O processo de obtenção da rede plana associada a um macrolugar, através da substituição da referência do macrolugar pela macro-rede associada apresenta-se na Figura 4-15(c) e resume-se aos seguintes passos:

- expande-se a área associada à representação do macrolugar e insere-se uma cópia da subpágina;
- procede-se à actualização das referências de todos os lugares e transições da macro-rede, de acordo com a ordem da instância da macro-rede a utilizar, garantindo a utilização de referências únicas a toda a especificação (no exemplo, dado ser a primeira instância da macro-rede *Seq*, as referências *pa*, *pb* e *ta* foram alteradas para *pa1*, *pb1* e *ta1*; notações alternativas poderiam utilizar *pa:1*, *pb:1* e *ta:1* ou *pa[1]*, *pb[1]* e *ta[1]*);
- os arcos ligados ao macrolugar são ligados aos lugares-fronteira associados;
- apaga-se a representação do macrolugar.

Do exposto, fica claro que não faz sentido proceder à marcação de um macrolugar (mas apenas dos lugares contidos na macro-rede associada).

#### 4.3.1.b Utilizando macrotransições

Seguidamente, aborda-se a utilização de macrotransições. De forma dual à apresentada para o macrolugar, a macrotransição corresponde a um macronó em que todos os nós da fronteira são do tipo transição. Neste caso, no entanto, não basta considerar a substituição da macrotransição pela macro-rede associada, num processo de simples colagem como o verificado com o macrolugar.

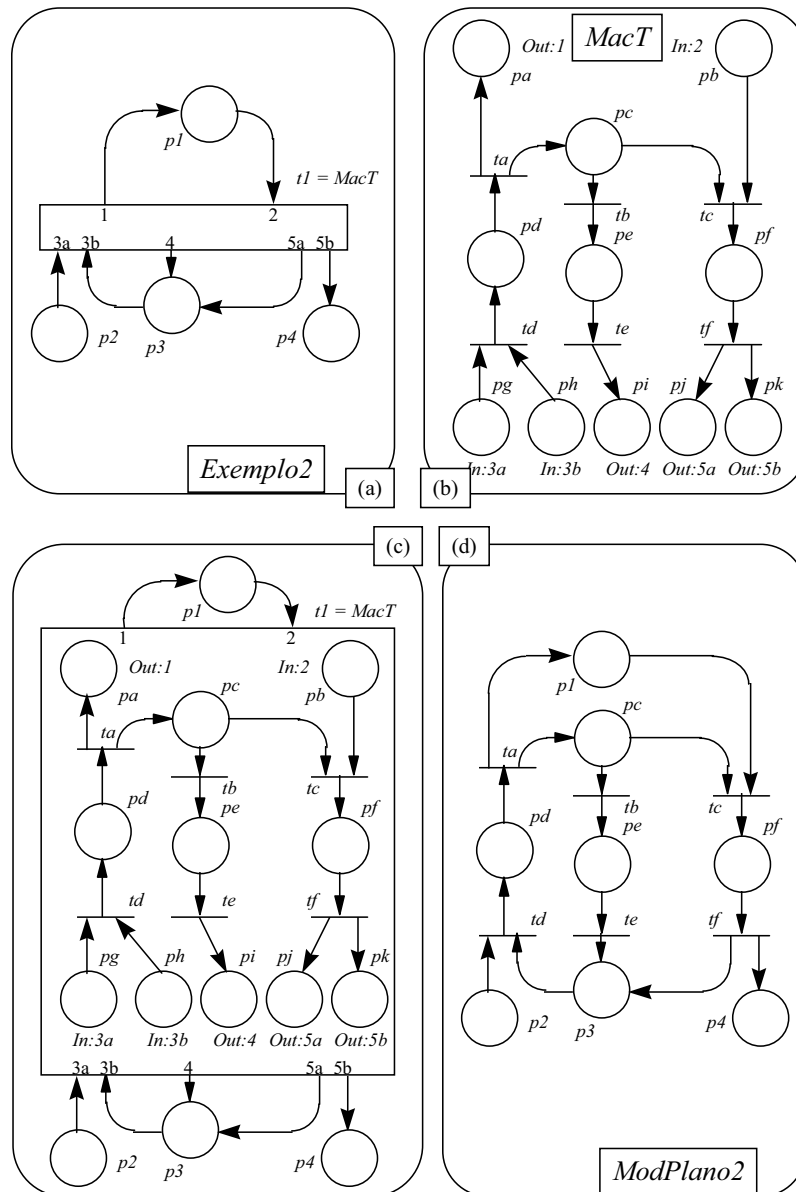


Figura 4-16 - Modelo recorrendo a uma macrotransição:

a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente.

Dado serem utilizadas RdP Coloridas, importa preservar as inscrições associadas aos arcos de entrada e saída da macrotransição, para adequada avaliação pelas transições contidas na macro-rede, no sentido da determinação dos vínculos possíveis. Assim, na macro-rede associada a uma macrotransição, são representados os lugares-fronteira da superpágina, bem como os arcos e inscrições associadas.

A Figura 4-16 apresenta um exemplo básico em que se pretende modelar a utilização de uma macrotransição. As inscrições associadas aos arcos do interface são representadas unicamente dentro da macro-rede, dado serem aí utilizadas para a determinação dos vínculos; na Figura 4-16 não são

explicitamente representadas, dado não sofrerem alterações devidas ao mecanismo de decomposição/agregação.

Na Figura 4-16(a) apresenta-se a página de topo da representação, incluindo uma referência à macrotransição  $t_1$ , implementado através da subpágina *MacT*.

De modo semelhante ao utilizado na subsecção anterior, também na notação gráfica aqui utilizada se realiza a indicação explícita das referências dos lugares-fronteira utilizados pelos arcos de interligação. Como será detalhado em subsecção posterior, as relações de sincronismo impostas pela ligação a uma mesma transição são identificadas através da utilização de um identificador constituído por um número (que é igual para os arcos ligados a essa transição) e por uma parte textual distinta (que permitirá a sua identificação).

O processo de obtenção da rede plana associada a uma macrotransição, através da substituição da referência da macrotransição pela macro-rede associada, ilustrado na Figura 4-16(c), resume-se aos seguintes passos:

- expande-se a área associada à representação da macrotransição e insere-se uma cópia da subpágina;
- procede-se à actualização das referências de todos os lugares e transições da macro-rede, de acordo com a ordem da instância da macro-rede a utilizar, garantindo a utilização de referências únicas a toda a especificação;
- os lugares-fronteira da macro-rede (bem como os arcos a eles ligados) são fundidos com os lugares correspondentes da superpágina (ou lugares-fronteira de macrolugares ou macroblocos quando interligados a outros macronós); as inscrições a utilizar nos arcos-fronteira são as referidas na macro-rede;
- apaga-se a representação da macrotransição.

No que se refere à marcação inicial da rede, esta deve ser realizada nos lugares presentes na superpágina, interligados à macrotransição (e não nos lugares-fronteira da macro-rede).

#### 4.3.1.c Utilizando macroblocos

Seguidamente, aborda-se a utilização de macroblocos. De forma semelhante à utilizada nas duas subsecções anteriores, a modelação da parte não-autónoma da rede não é considerada, dado ser independente dos mecanismos de estruturação apresentados.

O interface da macro-rede associada a um macrobloco é constituído por transições e lugares, de entrada e saída.

De modo resumido, devem-se aplicar os procedimentos indicados nas subsecções anteriores relativos à utilização de um macrolugar, para os lugares-porto, bem como os procedimentos indicados relativos à macrotransição, para as transições-porto.

A Figura 4-17 apresenta um exemplo básico utilizando um macrobloco. No exemplo apenas se pretende ilustrar a semântica associada à sua utilização; o realçe para a sua utilização em situações de modelação de sistemas será realizado em subsecção abordando um exemplo de aplicação.

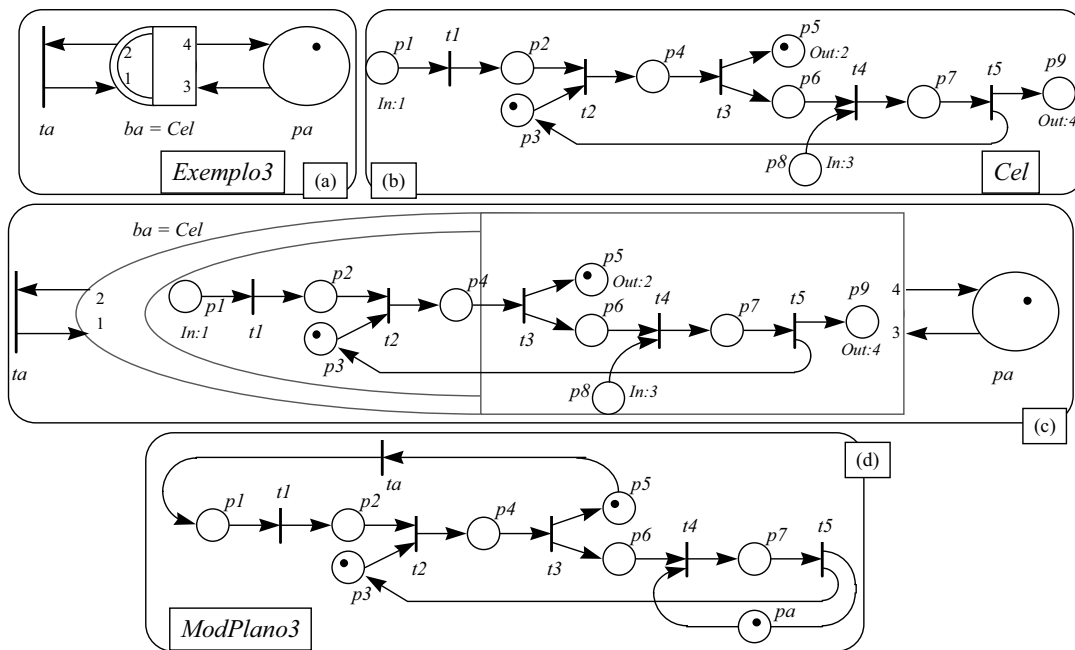


Figura 4-17 - Utilização de um macrobloco:

a) superpágina; b) subpágina; c) processo de junção; d) RdP plana equivalente.

Na Figura 4-17(a) apresenta-se a página de topo da representação, incluindo uma referência ao macrobloco  $ba$ , implementado através da subpágina  $Cel$ , representada na Figura 4-17(b).

O processo de obtenção da rede plana associada a um macrobloco, através da substituição da referência do macrobloco pela macro-rede associada apresenta-se na Figura 4-17(c) e resume-se nos seguintes passos:

- expande-se a área associada à representação do macrobloco e insere-se uma cópia da subpágina;

- procede-se à actualização das referências de todos os lugares e transições da macro-rede, de acordo com a ordem da instância da macro-rede a utilizar, garantindo a utilização de referências únicas a toda a especificação;
- os arcos ligados ao macrobloco são ligados aos portos associados de acordo com os procedimentos atrás apresentados para os macrolugares e macrotransições, dependendo do tipo de interface;
- apaga-se a representação do macrobloco.

#### 4.3.1.d *Sobre as macro-redes*

De modo informal, uma macro-rede pode ser caracterizada como uma página comum que contem um conjunto de nós especiais, responsáveis pelo seu interface com a superpágina. Esse conjunto de nós é constituído por lugares-fronteira.

Os lugares-fronteira são identificados com as seguintes três características:

- a primeira pode tomar os valores “In”, “Out” ou “In/Out” e refere-se ao sentido das marcas que “circulam” nos arcos ligados ao macronó;
- a segunda é um número de ordem sequencial do porto (alternativamente pode ser utilizado o próprio identificador do lugar, como referido em subsecção anterior e dependendo do ambiente gráfico de suporte à especificação); para lugares-fronteira de macro-redes associadas a macrotransições, o número de referência é comum a todos os lugares-fronteira interligados a uma mesma transição (desta forma, para as macro-redes associadas a macrotransições não é válido interligar um lugar-fronteira a mais do que uma transição da macro-rede; isto não constitui uma limitação às capacidades de modelação, dado que quando for necessário modelar a ligação de um lugar na superpágina a N transições da macro-rede, podem ser utilizados N lugares-fronteira na macro-rede que serão fundidos com o lugar da superpágina);
- a terceira é uma letra, utilizável apenas em lugares-fronteira de macro-redes, permitindo distinguir entre as várias ligações a uma mesma transição.

Considerando o caso de um macrolugar:

- o atributo “In” é utilizado para um lugar-fronteira que, do exterior, só receba arcos de entrada;

- o atributo “Out” é utilizado quando o interface com o exterior é garantido unicamente através de arcos de saída;
- o atributo “In/Out” é utilizado quando a ligação ao exterior inclua arcos de entrada e saída.

Considerando o caso de uma macrotransição:

- o atributo “In” é utilizado com um lugar-fronteira que representa um lugar do exterior que gera um arco de entrada na macrotransição;
- o atributo “Out” é utilizado com um lugar-fronteira que representa um lugar do exterior que recebe um arco de saída da macrotransição;
- o atributo “In/Out” é utilizado quando a ligação ao lugar exterior representado inclua arcos de entrada e saída.

A possibilidade de interligação directa entre macronós é uma característica necessária para a utilização sistemática deste mecanismo de decomposição modular de RdP. A sua implementação necessita, no entanto, de uma caracterização mais detalhada sobre a natureza dos lugares-fronteira utilizados nos macronós.

Deste modo, enquanto os lugares-fronteira presentes num macrolugar representam lugares com existência em termos do modelo plano, os lugares-fronteira presentes numa macrotransição representam os lugares exteriores à macrotransição e a sua representação dentro da macro-rede apenas se deve à necessidade de coerência associada à utilização das capacidades de compactação intrínsecas às RdP Coloridas. Isto é, os lugares-fronteira de uma macrotransição são “virtuais”, no sentido em que representam apenas referências para os lugares reais, e não nós efectivos presentes no modelo plano.

Deste modo, os lugares-fronteira podem ser divididos em dois tipos: os reais (que pertencem ao modelo plano) e os virtuais (que representam os lugares “reais”, identificados na superpágina).

No que respeita aos lugares-fronteira utilizados pelos macroblocos, estes podem corresponder a lugares “reais” ou a lugares “virtuais”, dependendo da sua natureza (isto é, se modelam a ligação a macrotransições ou a macrolugares), de acordo com o exposto.

De realçar que a caracterização referida permite a utilização de um macronó como nó-fronteira de outros macronós (desde que, obviamente, se garantam as características de grafo bipartido das RdP); a obtenção do modelo plano equivalente realiza-se de acordo com os procedimentos expostos.

#### 4.3.1.e Explorando a utilização de macronós

Os mecanismos de decomposição horizontal propostos podem ser considerados como meros expedientes predominantemente gráficos para a estruturação do modelo.

No entanto, as actividades de modelação de sistemas, como actividades criativas que são, necessitam de ter um amplo leque de técnicas de suporte à modelação, suficientemente flexíveis e simultaneamente fiáveis, tarefa para a qual o presente trabalho tenta ser um contributo e que justifica a classificação das técnicas de decomposição horizontal como de estruturação hierárquica do modelo.

Na presente subsecção procede-se à análise de um exemplo inicialmente apresentado em [Gomes & Steiger-Garção, 92] e retomado em [Gomes & Steiger-Garção, 95d]. Refere-se a um sistema com  $N$  células ligadas em cascata (com  $N=3$ ).

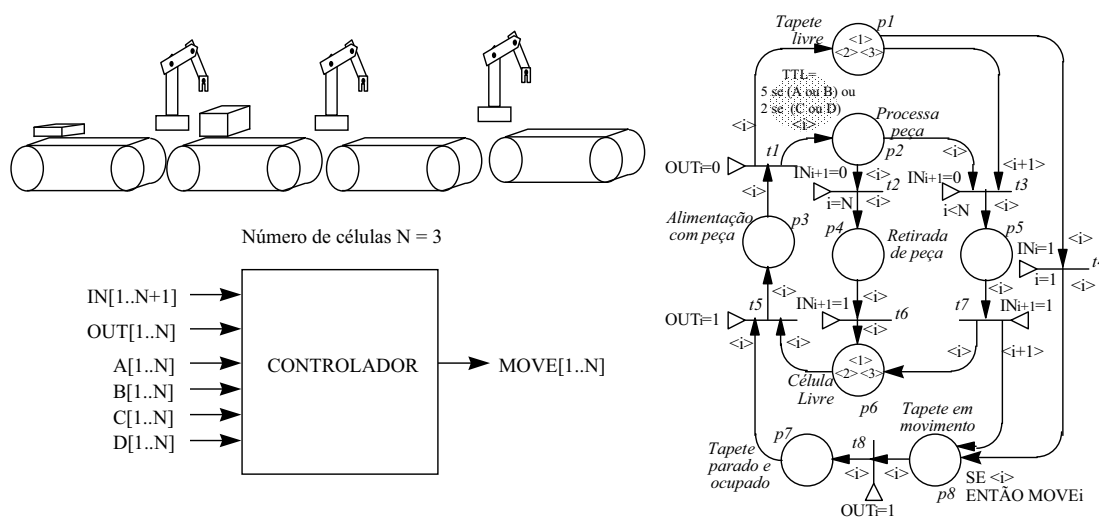


Figura 4-18 - Modelação de um sistema de produção em linha.

Cada uma das células é composta por um centro de maquinagem ou *robot* e por um tapete de transporte. Os tapetes de transporte dispõem de sensores para detecção de peças/“palletes” nas suas entradas e saídas, que fornecem os sinais  $IN[1..N]$  e  $OUT[1..N]$ , respectivamente; comportam-se como armazéns (“buffers”) limitados, podendo conter no máximo uma peça à entrada e outra à saída. O seu movimento é controlado através dos sinais  $MOVE[1..N]$ . Cada célula dispõe de um sistema de identificação de peças, fornecendo quatro saídas ( $A[1..N]$ ,  $B[1..N]$ ,  $C[1..N]$  e  $D[1..N]$ ); os tempos de processamento das várias peças não são uniformes.

A saída de peças/“palletes” é realizada através de um tapete de saída, considerado como armazém com capacidade infinita e fornecendo o sinal  $IN[N+1]$  de detecção de entrada ocupada.

A Figura 4-18 apresenta a caracterização simplificada do sistema, bem como a RdP-R de controlo do movimento dos tapetes, apresentado em [Gomes & Steiger-Garção, 92].

Devido à utilização das características das RdP Coloridas, o modelo apresentado na Figura 4-18 é facilmente alterado para representar o controlo para sistemas com N células, bastando para tal alterar a marcação inicial dos lugares p1 e p6, utilizando em cada um dos lugares tantas marcas quanto os andares da linha. A “regularidade” do modelo é quebrada com a utilização da transição t4 responsável por modelar a admissão de uma nova peça ou “pallette” na linha (fluxo vertical representado à direita da figura).

É possível representar o mesmo sistema através de um modelo mais abstracto (ver Figura 4-19), em que, para simplificação gráfica, a componente não-autónoma não é representada e onde a transição t4 e os lugares p1 e p6 da Figura 4-18 foram mantidos (através da transição tx e lugares px e py, respectivamente). Recorre-se à utilização de um macrolugar (representando a sequência associada a “tapete ocupado” e constituída pelos lugares p7 e p8) e uma macrotransição (representando a dinâmica associada à operação da célula).

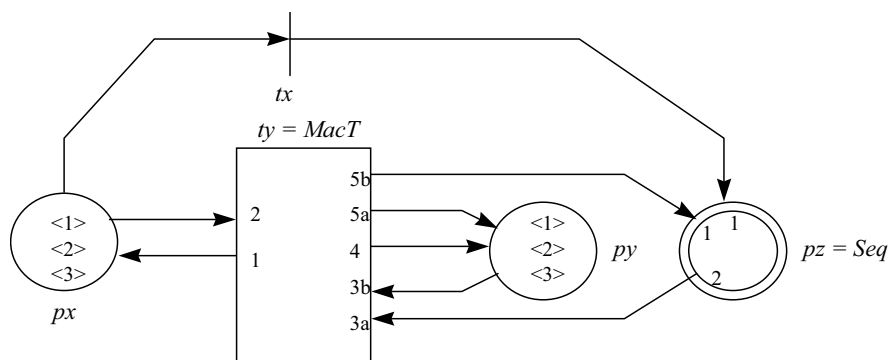


Figura 4-19 - Modelação modular do sistema de produção em linha.

As macro-redes associadas foram apresentadas anteriormente (ver Figura 4-15 e Figura 4-16).

No sentido de ilustrar os ganhos de compactação associados à utilização de modelos coloridos, apresenta-se na Figura 4-20 uma RdP de baixo-nível que modela o mesmo sistema de controlo. Para não tornar o modelo muito “pesado”, omitem-se os identificadores dos lugares e transições. Na Figura 4-20, identificam-se várias regiões:

- três regiões, representadas em camadas e associadas à utilização das marcas coloridas <1>, <2> e <3>, respectivamente;
- várias regiões verticais onde se podem estabelecer relações entre os lugares e transições dessa região vertical e os lugares e transições equivalentes utilizados no modelo da Figura

4-18 e referidos no topo da Figura 4-20, e com os lugares e transições equivalentes utilizados no modelo da Figura 4-17 e referidos na base da Figura 4-20.

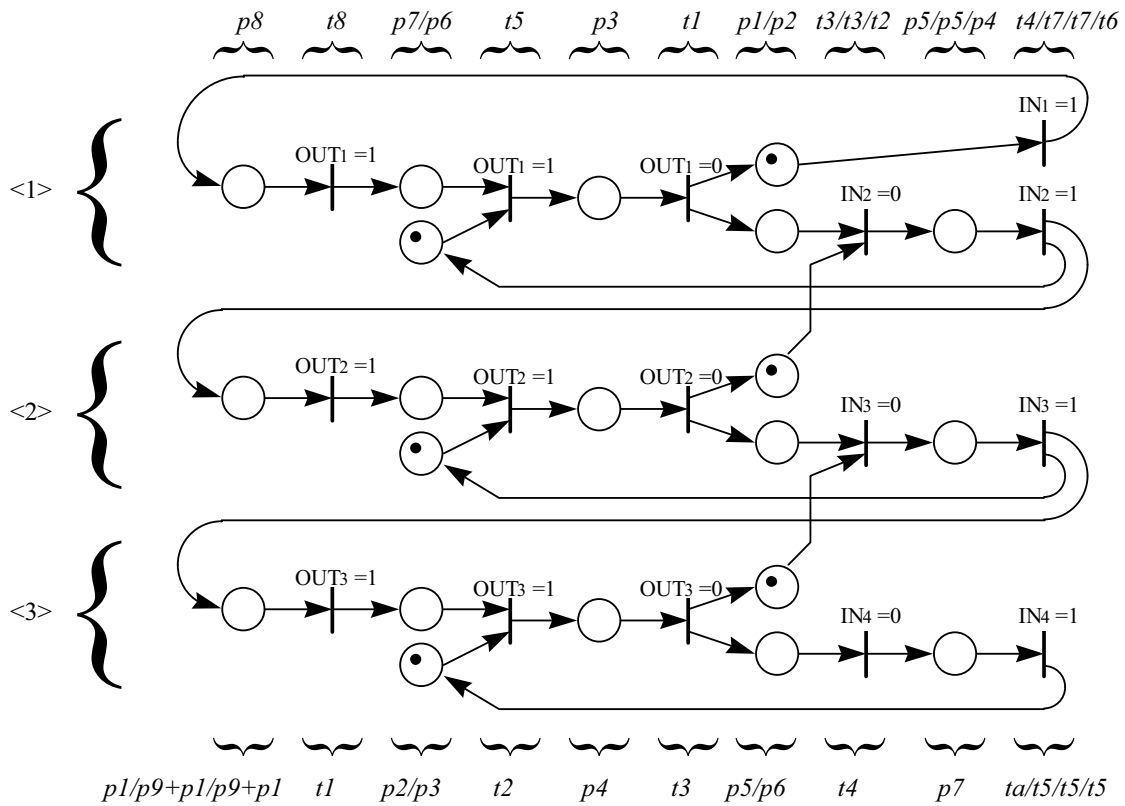


Figura 4-20 - Modelo de baixo-nível do sistema de produção em linha.

Cada uma das regiões horizontais contendo a sub-rede associada ao controlo de cada um dos andares do sistema é caracterizável através de um bloco. A sua representação foi apresentada na Figura 4-17.

Torna-se, deste modo, natural representar o modelo do sistema baseado em blocos, como apresentado na Figura 4-21.

No sentido de garantir a re-utilização da macro-rede associada ao macrobloco, considera-se que, quando a um porto do interface não se encontra ligado nenhum arco, a obtenção do modelo plano equivalente continua a ser possível, não sendo, simplesmente, o porto associado utilizado; este pressuposto permite a modelação do terceiro andar como um bloco semelhante aos restantes.

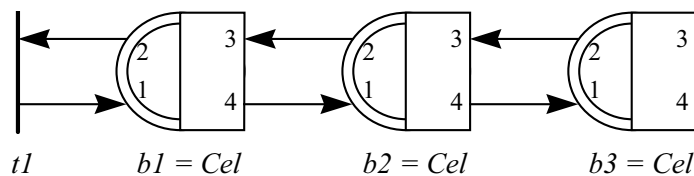


Figura 4-21 - Modelo utilizando blocos do sistema de produção em linha.

Evidencia-se o facto do modelo do sistema reflectir directamente a organização lógica da planta de fábrica, de modo tal que, se a linha passar a conter um número  $N$  (maior que três) de células, bastará aumentar o número de blocos interligados em cascata proporcionalmente.

Esta é uma potencialidade de modelação tornada disponível pela utilização do conceito proposto de macrobloco.

### 4.3.2 Representação nodal estruturada através de vectores

Importa evidenciar o interesse em utilizar uma notação gráfica que suporte uma representação mais compacta de modelos em que, como na Figura 4-21, a repetição de nós ou sub-redes seja uma característica facilmente identificável. Esta é, aliás, uma das motivações genuínas para a utilização de RdP Coloridas; a solução aí encontrada utiliza estruturas de dados associadas às marcas para permitir a paginação compactada das sub-redes modulares.

Em linha com a decomposição modular utilizada neste capítulo, introduz-se, seguidamente, a estruturação do modelo associada aos nós do grafo. É uma técnica comum em áreas de aplicação como a da especificação de sistemas digitais e computacionais, em que a utilização de registos, barramentos de interligação e unidades modulares torna natural a utilização de vectores para representar, de modo compactado, um conjunto de entidades semelhantes cumprindo uma mesma função.

Apresenta-se na Figura 4-22 um modelo utilizando um vector de blocos  $Cel$  com três elementos, que é equivalente aos modelos atrás apresentados para representar o controlo do sistema de produção. As anotações dos arcos serão justificadas na subsecção seguinte.

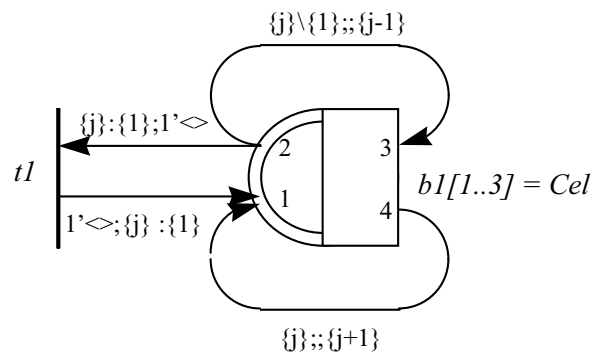


Figura 4-22 - Modelo utilizando uma estrutura de blocos do sistema de produção em linha.

### 4.3.2.a Semântica associada

De uma forma geral, a representação estruturada através de vectores pode ser associada a qualquer nó da representação. Neste contexto, por nó entende-se, para além dos lugares e transições, os macronós apresentados, bem como os metanós e supernós a apresentar em subsecções posteriores. As estruturas consideradas incluem os vectores com N dimensões. Como caso de particular interesse, referem-se os vectores unidimensionais, de que a Figura 4-22 é um exemplo de aplicação.

A notação gráfica utilizada para representar estruturas de nós é a seguinte:

identificador\_de\_nó [índiceA\_inicial .. índiceA\_final]...[índiceN\_inicial .. índiceN\_final]

em que:

- índiceA\_inicial, índiceN\_inicial, índiceA\_final e índiceN\_final representam números inteiros quaisquer (diferentes);
- a expressão representa (kA \* ... \* kN) instâncias de nós do tipo indicado, com  $kX = |\text{índiceX\_inicial} - \text{índiceX\_final}| + 1$
- cada instância é representada através de identificador\_de\_nó [índiceA]...[índiceN] com índiceX variando entre índiceX\_inicial e índiceX\_final;
- X varia entre A e N.

Na caracterização que se segue utilizam-se vectores unidimensionais para simplificar a apresentação; no entanto, a generalização a vectores multidimensionais não apresenta complexidade adicional.

No caso particular dos nós que possuem uma marcação associada, isto é, dos lugares e dos metalugares (a apresentar posteriormente), importa possibilitar a utilização de estados de marcação inicial não uniformes para todos os nós do vector, bem como identificar a notação utilizada. Assim, caso a marcação inicial seja uniforme para todos os lugares do vector, a sua indicação não sofre alterações em relação ao normalmente utilizado. Caso se pretenda uma marcação inicial não uniforme, a sua indicação é realizada através de uma expressão complexa com o seguinte formato:

{expressão\_de\_índice}<marca>...{expressão\_de\_índice}<marca>

em que marca se refere a uma marca colorida comum, nos atributos da qual também podem ser utilizadas variáveis referidas na expressão\_de\_índice.

No que se refere a *expressão\_de\_índice* consideram-se as seguintes expressões válidas:

- caso I: {expressão}
- caso II: {expressão} \ {excepção1, ..., excepçãoN}
- caso III: {expressão} : {extensão1, ..., extensãoN}

O caso I refere-se à situação mais geral; são válidas todas as expressões que se avaliem como um inteiro dentro do intervalo associado ao índice do nó vectorizado associado.

O caso II permite excluir as situações em que “*expressão*” seja avaliada como “*excepçãoX*”, enquanto o caso III permite considerar unicamente as situações em que “*expressão*” seja avaliada como “*extensãoX*”.

Considerem-se os seguintes exemplos de expressões de marcação inicial associadas a um lugar vectorizado com três lugares  $p1[1..3]$  :

- $2' \diamond$  equivalente à seguinte marcação uniforme  $p1[1] \leftarrow 2' \diamond$ ;  $p1[2] \leftarrow 2' \diamond$ ;  $p1[3] \leftarrow 2' \diamond$ ;
- $\{j\} \setminus \{3\} \langle 'a', j \rangle \{j\} : \{3\} \langle 'b', j \rangle$  equivalente à seguinte marcação  $p1[1] \leftarrow \langle 'a', 1 \rangle$ ;  $p1[2] \leftarrow \langle 'a', 2 \rangle$ ;  $p1[3] \leftarrow \langle 'b', 3 \rangle$ .

Para o caso particular das transições, importa possibilitar a utilização de guardas e eventos associados não uniformes para todas as transições do vector, bem como identificar a notação utilizada. Para os casos em que as guardas ou os eventos associados sejam uniformes para todas as transições do vector, a sua indicação não sofre alterações em relação ao normalmente utilizado.

Para os casos em que se pretenda uma guarda não uniforme, a sua indicação é realizada através de uma expressão complexa com o seguinte formato:

$$\{\text{expressão\_de\_índice}\} \text{guarda} \dots \{\text{expressão\_de\_índice}\} \text{guarda}$$

em que *guarda* se refere a uma guarda colorida comum, na qual também podem ser utilizadas como atributos variáveis utilizadas na *expressão\_de\_índice*.

Para os casos em que se pretenda um evento não uniforme para o vector de transições, a sua indicação é realizada através de uma expressão complexa com o seguinte formato:

$$\{\text{expressão\_de\_índice}\} \text{evento} \dots \{\text{expressão\_de\_índice}\} \text{evento}$$

em que *evento* se refere a um evento comum, na qual também podem ser utilizadas variáveis referidas na *expressão\_de\_índice*.

No que se refere a *expressão\_de\_índice* consideram-se válidas as expressões caracterizadas anteriormente.

A utilização de conjuntos de nós representados graficamente apenas por uma instância desse tipo de nós, traz novas necessidades em termos da expressividade das expressões associadas aos arcos e necessidade de definição da forma utilizada para a sua avaliação.

No que se segue, as expressões associadas aos arcos são avaliadas localmente, em termos de três entidades: a expressão associada ao arco, o nó de origem e o nó de destino.

Desta forma, excluem-se as avaliações conjuntas de expressões de arcos, envolvendo vários arcos com vínculos comuns; a avaliação das expressões associada à representação estruturada através de vectores afasta-se, pois, significativamente da avaliação realizada para as RdP Coloridas, onde os vínculos possíveis associados às transições estão no centro da interpretação. Essa a razão porque, na Figura 4-22, por exemplo, as variáveis  $j$  utilizadas nas expressões de ambos os arcos ligados à transição  $t_1$ , não são correlacionadas e necessitam de ser especificadas independentemente; no caso, a restrição da variável a  $j=1$  é realizada em ambos os arcos e não apenas num deles (numa transição colorida simples, a instanciação da variável  $j=1$  num dos arcos, impõe essa restrição aos vínculos possíveis para essa transição).

Cada uma das três entidades acima referidas (expressão associada ao arco, nó de origem e nó de destino) é classificada em uma de duas situações: simples ou contendo referências a estruturas.

Desta forma, o espaço de casos a caracterizar divide-se em oito regiões, de acordo com a Tabela 4-1. A análise das semânticas associadas a cada uma das regiões é feita seguidamente, em termos das traduções das representações estruturadas em representações não-estruturadas equivalentes. As marcas eventualmente existentes nos nós-vector não são explicitamente representadas, aplicando-se a notação atrás proposta.

|                 |            |            |            |            |
|-----------------|------------|------------|------------|------------|
| Nó de origem    | <b>com</b> | <b>com</b> | <b>sem</b> | <b>sem</b> |
| Arco <b>com</b> | I          | III        | V          | VII        |
| Arco <b>sem</b> | II         | IV         | VI         | VIII       |
| Nó de destino   | <b>com</b> | <b>sem</b> | <b>com</b> | <b>sem</b> |

Tabela 4-1 - Caracterização das situações a considerar,

em termos da dependência em relação à representação estruturada através de vectores.

As regiões VII e VIII não serão, obviamente, analisadas; a região VIII corresponde ao modelo comum sem qualquer utilização de representação estruturada através de vectores e a região VII corresponde à utilização de referências a estruturas nas expressões dos arcos sem que qualquer dos nós envolvidos as utilize, situação em que o arco também se considerará normal (isto é, não é válida a utilização de características estruturadas em arcos não interligados a pelo menos um nó vectorizado).

Inicia-se a análise pelas regiões pares, em particular pela região IV, em que se está em presença de um arco com uma inscrição colorida simples, um nó de origem vectorizado e um nó de destino simples. O arco referido representa um conjunto de  $N$  arcos interligando cada um dos nós do vector de origem e o nó de destino, em que a expressão do arco inicial se repete em todos os arcos depois de descompactado. A Figura 4-23 resume a semântica utilizada.

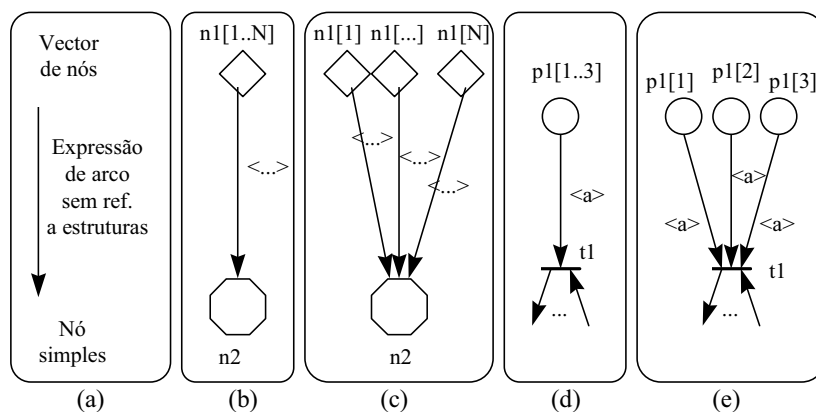


Figura 4-23 - Semântica associada à região IV da Tabela 4-1.

Na Figura 4-23(a) caracteriza-se o caso geral endereçado, enquanto a Figura 4-23(b) apresenta a representação gráfica estruturada através de vectores e a Figura 4-23(c) a rede não estruturada equivalente. São utilizados losangos e octogonos para representar qualquer par de nós em que um arco possa ser utilizado. As Figura 4-23(d) e Figura 4-23(e) apresentam um exemplo recorrendo a um vector de lugares e uma transição.

A região VI é, de certa forma, complementar à anterior. Aqui, analisa-se a situação de um arco com uma inscrição colorida simples, um nó de origem simples e um nó de destino vectorizado. O arco referido representa um conjunto de  $N$  arcos interligando o nó de origem a cada um dos nós do vector de destino; a expressão de arco utilizada na representação estruturada através de vectores repete-se em todos os arcos da representação descompactada. A Figura 4-24 resume a semântica utilizada.

A Figura 4-24(a) caracteriza o caso geral em análise, enquanto a Figura 4-24(b) apresenta a representação gráfica estruturada através de vectores e a Figura 4-24(c) a rede não estruturada equivalente. As Figura 4-24(d) e Figura 4-24(e) apresentam um exemplo recorrendo a um vector de transições e um lugar.

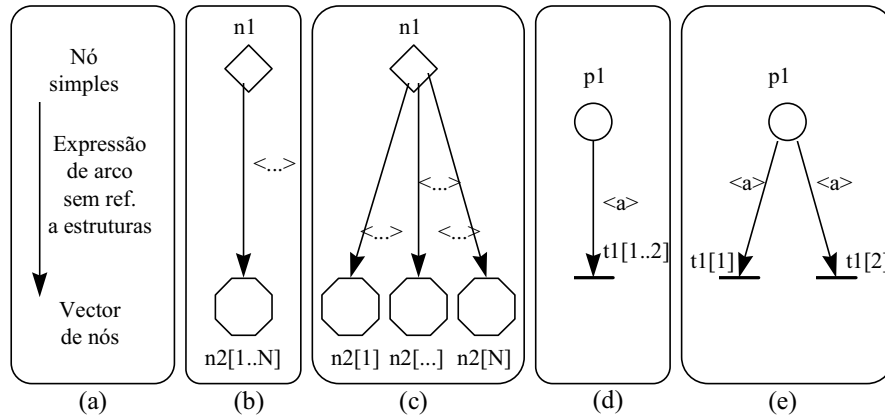


Figura 4-24 - Semântica associada à região VI da Tabela 4-1.

A região II da Tabela 4-1 corresponde à junção das características das regiões IV e VI da mesma tabela. Considerando um nó de origem com N instâncias e um nó de destino com M instâncias, o arco que interliga os dois vectores de nós representa um conjunto de N\*M arcos interligando cada uma das instâncias do nó de origem a cada uma das instâncias do nó de chegada. Novamente, a expressão de arco utilizada na representação estruturada através de vectores repete-se em todos os arcos da representação descompactada. A Figura 4-25 resume a semântica utilizada.

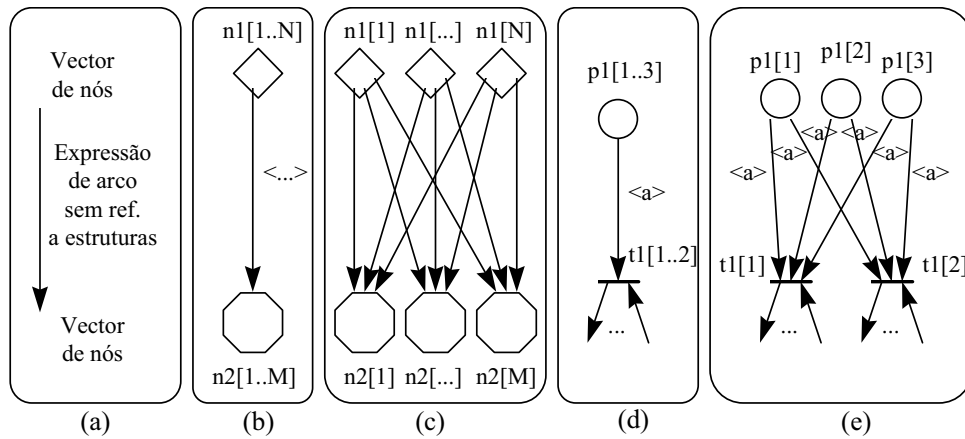


Figura 4-25 - Semântica associada à região II da Tabela 4-1.

Na Figura 4-25(a) caracteriza-se o caso geral em análise, enquanto a Figura 4-25(b) apresenta a representação gráfica estruturada através de vectores e a Figura 4-25(c) a rede não estruturada equivalente. As Figura 4-25(d) e Figura 4-25(e) apresentam um exemplo elementar.

As restantes três regiões da Tabela 4-1 (I, III e V) referem-se a arcos em que a inscrição utilizada, para além da expressão comum em RdP-R, possui mais dois atributos, permitindo indicar os índices associados às instâncias de nós vectorizados em relação aos quais são realizadas ligações, através dos arcos “descompactados”.

A semântica associada a estas três regiões (I, III e V) é a resultante da utilização concomitante do exposto em relação às três regiões próximas (II, IV e VI, respectivamente) com a utilização de filtros de selecção ou anulação dos arcos gerados pela avaliação dos novos dois atributos.

As expressões de arcos associadas aos três casos em estudo têm os seguintes formatos:

- região I:  
índice\_no\_vector\_de\_origem ; inscrição; índice\_no\_vector\_de\_destino
- região III:  
índice\_no\_vector\_de\_origem ; inscrição
- região V:  
inscrição; índice\_no\_vector\_de\_destino

No que se refere à expressão associada a “inscrição”, trata-se de uma inscrição de arco comum, no sentido utilizado nas RdP Coloridas e nas RdP-R, havendo a acrescentar a possibilidade de se utilizarem variáveis também utilizadas pelas expressões de índice, onde se efectua a sua instanciação.

No que se refere às expressões “*índice\_no\_vector\_de\_origem*” e “*índice\_no\_vector\_de\_destino*” aplica-se o que atrás foi dito em relação à “*expressão\_de\_índice*”, quando das considerações sobre a marcação inicial do vector de lugares.

Considerem-se como exemplos as expressões utilizadas na Figura 4-22, em que se utiliza o nó  $b1[1..3]$ :

- $\{j\};\{1\};1' \diamond$  (região III, caso III) considera um arco com origem no nó de índice 1 e com a inscrição associada  $1' \diamond$
- $1' \diamond; \{j\};\{1\}$  (região V, caso III) considera um arco cujo nó de destino possui o índice 1 e com uma inscrição associada  $1' \diamond$
- $\{j\} \setminus \{1\}; \{j-1\}$  (região I, casos II e I) considera dois arcos com origem nos nós de índices 2 e 3, com destino para os nós de índices 1 e 2, respectivamente e sem inscrições (isto é, transportando marcas de baixo nível);
- $\{j\}; \{j+1\}$  (região I, caso I) considera dois arcos com origem nos nós de índices 1 e 2, com destino para os nós de índice 2 e 3, respectivamente e sem inscrições.

#### 4.3.2.b Tradução em RdP Coloridas

Importa realçar que a introdução da capacidade de representação estruturada através de vectores não corresponde à criação de novas capacidades de modelação em relação às apresentadas para as RdP-R (ou para as RdP Coloridas, que são utilizadas como classe de base do modelo proposto para as características autónomas). Isso ficou demonstrado quando se apresentaram as técnicas de tradução de uma representação utilizando características estruturadas numa outra que as não utiliza.

Todavia, a utilização da representação estruturada através de vectores é plenamente justificada; o desenhador do sistema disporá de mais técnicas para proceder à sua modelação, podendo seleccionar a que corresponder à forma mais simples e legível para um determinado sistema.

É possível obter representações com níveis de compactação gráfica intermédias em relação aos níveis extremos apresentados (representação estruturada através de vectores e representação não estruturada equivalente); a Figura 4-26 dá mostras dessa flexibilidade.

É trivial identificar um conjunto de procedimentos de tradução de algumas representações vectoriais em atributos de cor das marcas utilizadas. Não se desenvolve seguidamente a análise destes procedimentos de tradução, dado serem inconsequentes em função do presente trabalho.

A afirmação atrás produzida de que as expressões associadas aos arcos são avaliadas localmente, impedindo, por exemplo, a avaliação simultânea de um grupo de arcos envolvidos numa transição, é ilustrada na Figura 4-26.

Na Figura 4-26(a) apresenta-se uma notação não permitida para modelar um conjunto de  $N$  lugares e lugares complementares; paralelamente, nas Figura 4-26(b) e Figura 4-26(c) são apresentadas duas representações correctas comportamentalmente equivalentes, uma recorrendo a nós-vector e outra a RdP Coloridas.

De notar que, as cores associadas aos lugares das Figura 4-26(b) e Figura 4-26(c) não são as mesmas; na Figura 4-26(c) é necessário utilizar uma cor com mais um atributo em relação à Figura 4-26(b), com capacidade de armazenar a ordem do lugar.

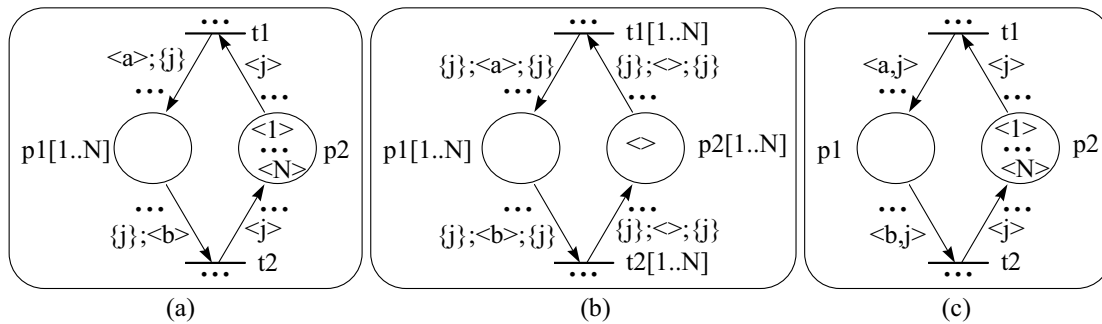


Figura 4-26 - Utilização de características estruturadas através de vectores em modelos comportamentalmente equivalentes: a) representação não permitida; b) representação estruturada válida; c) representação colorida válida.

### 4.3.3 Decomposição vertical

A decomposição vertical tem uma natureza substancialmente diferente da decomposição horizontal. Representa uma dependência hierárquica entre o nó do nível superior e uma subpágina autónoma contida no nível inferior, no sentido em que o nó do nível superior representa de forma autónoma o submodelo associado.

Desta forma é válido proceder à análise, simulação ou execução do modelo ao nível da superpágina, independentemente das subpáginas dependentes.

Esta forma de decomposição pode ser associada à utilização de diversos tipos de mecanismos de abstracção, presentes em várias linguagens de modelação de sistemas.

Como primeira utilização, refira-se o suporte às simulação e implementação multinível do sistema, isto é, partindo de uma especificação de um sistema utilizando uma abordagem descendente (constituída por  $N$  níveis de abstracção) e dado um nível de detalhe pretendido, é possível obter o modelo associado e proceder à sua execução. Como exemplo da utilização desta capacidade de modelação multinível, refira-se a linguagem VHDL (Very high speed integrated circuit Hardware Description Language) [Shahdad et al., 85], em que a descrição de um sistema electrónico digital pode ser realizada a vários níveis, baseando-se nas denominadas “entidades de projecto”, constituídas por um “interface” e por uma ou mais partes denominadas “corpos”. Cada corpo pode implementar uma vista diferente sobre as entidades de projecto, nomeadamente comportamental e estrutural.

Também na área das linguagens de programação, outra aplicação típica em que se utiliza a capacidade de encapsulamento multinível, é a utilização de subprogramas. Dependendo da existência ou não de parâmetros, a modelação de subprogramas em RdP-R pode ser realizada de formas diversas, como mais à frente será apresentado.

São consideradas duas possibilidades para proceder a uma decomposição de tipo vertical:

- utilizando metalugares;
- utilizando metatransições.

Quer os metalugares, quer as metatransições podem ser referidos por metanós. Consideram-se dois tipos de metanós: simples e com história. Posteriormente, retomar-se-á esta classificação.

A representação dos metanós possui duas componentes: a primeira a utilizar na superpágina, e é comportamentalmente semelhante a um nó do mesmo tipo do metanó; a segunda corresponde à vista detalhada, associada à subpágina e é constituída por uma sub-rede autónoma, denominada meta-rede. A marcação inicial da meta-rede é determinada pelos vínculos dos arcos de entrada no metanó.

Ao nível da superpágina, o metalugar comporta-se como um lugar ordinário, interligando-se através de arcos a nós com semântica de transição; pode ser marcado com marcas de uma determinada cor, bem como habilitar transições de saída a ele ligadas. As marcas que pode armazenar, denominadas por metamarcas, representam, para além do estado de marcação da superpágina, a actividade da meta-rede autónoma associada.

Por seu lado, ao nível da superpágina, a metatransição comporta-se, na sua forma mais simples, como uma transição ordinária à qual se associa um evento não-instantâneo, isto é, como uma transição comportamentalmente equivalente a uma sub-rede constituída por uma transição recebendo os arcos de entrada e outra a que se ligam os arcos de saída, interligadas através de um metalugar. À transição de entrada (saída) encontra-se associado um evento de início (fim, respectivamente) de actividade. No caso mais geral, que se apresentará posteriormente, são consideradas várias alternativas para conclusão da actividade.

Do ponto de vista da representação gráfica a inserir na superpágina, utilizam-se as seguintes convenções:

- um metalugar é representado como um lugar mas utilizando uma linha tracejada;
- uma metatransição é representada através de um rectângulo a tracejado.

A descrição detalhada dos mecanismos de decomposição vertical a realizar nas subsecções seguintes inicia-se pela caracterização dos metanós ao nível da superpágina, sendo complementada pela descrição da meta-rede e suas características, aplicáveis a ambos os tipos de metanós. Apresentam-se alguns exemplos e definições com o objectivo de esclarecer as semânticas associadas a cada uma das estruturas propostas de metanó.

### 4.3.3.a Utilizando metalugares

A Figura 4-27 apresenta a representação gráfica associada a um metalugar, bem como o modelo equivalente a utilizar para execução, simulação ou análise da rede onde o metalugar está inserido.

Para além do comportamento como um lugar vulgar, enquanto lugar inserido numa superpágina, o metalugar possui uma característica distintiva: a de ser marcado através de metamarcas.

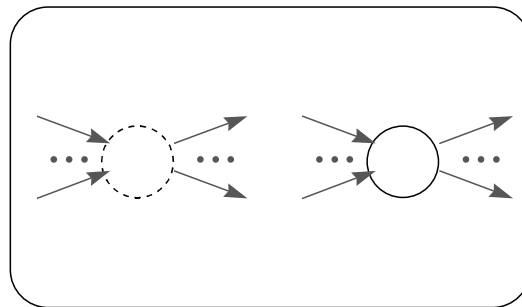


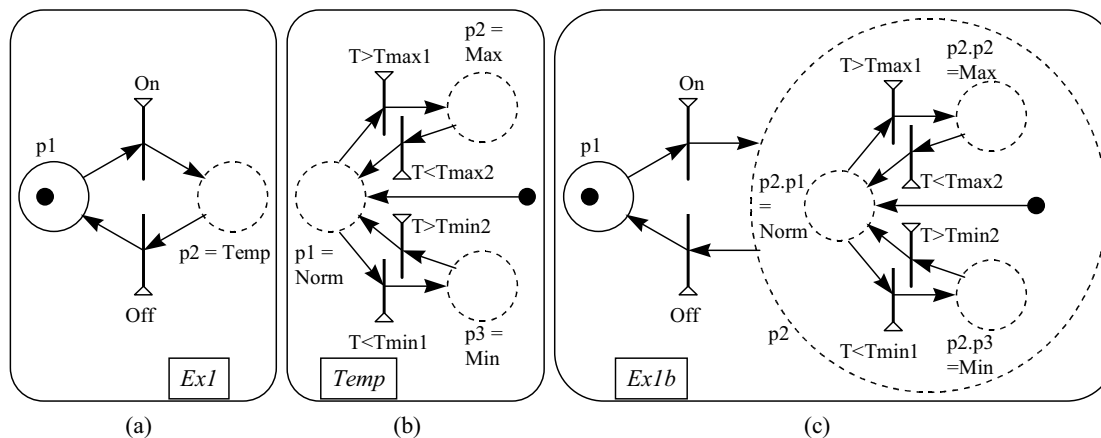
Figura 4-27 - Metalugar: representação gráfica e modelo associado na superpágina.

Uma metamarca possui várias componentes:

- uma estrutura de dados ao nível da superpágina, isto é, uma marca colorida no sentido das RdP Coloridas;
- uma sub-RdP-R associada;
- mecanismos de sincronização entre a criação (remoção) da metamarca na superpágina e a activação (desactivação, respectivamente) da RdP-R associada.

A Figura 4-28 apresenta um exemplo básico em que se pretende ilustrar a modelação hierárquica multinível de um sistema com vários modos de funcionamento. Considere-se, para isso, um sistema de aquecimento cujo controlo pode ser especificado a vários níveis de detalhe.

No nível mais elevado, representado na Figura 4-28(a), é modelado o comportamento biestável de ligar/desligar. O estado de ligado, modelado através do metalugar  $p_2$ , é detalhado através da meta-rede *Temp*, representada na Figura 4-28(b) e constituída por três metalugares, cada um dos quais é objecto de detalhe posterior (não representado na figura). O arco iniciado por uma círculo representa a marcação inicial da meta-rede, afectando o lugar  $p_1$  de *Temp* quando da marcação de  $p_2$  de *Ex1* (como nos statecharts). Corresponde a uma notação simplificada utilizável quando as marcas são de baixo-nível; em subsecção posterior retomar-se-á o tema para marcações coloridas.



(a) (b) (c)

Figura 4-28 - Modelo hierárquico parcial de controlo de aquecimento:

a) nível de topo; b) nível 2; c) visão planar da representação hierárquica.

A especificação poderá, a exemplo da notação utilizada nos statecharts, ser representada de forma compactada, utilizando a noção de área para melhorar a legibilidade intuitiva da representação (ver Figura 4-28(c)).

A sincronização entre a superpágina *Ex1* e a meta-rede *Temp* é realizada em dois instantes distintos:

- o primeiro quando da ocorrência do evento *On*, responsável pela criação simultânea de uma marca no lugar *p2* de *Ex1* e outra no lugar *p1* de *Temp* (ambas de baixo-nível);
- o segundo quando da ocorrência do evento *Off*, responsável pela remoção simultânea da marca no lugar *p2* de *Ex1*, da marca presente num dos metalugares de *Temp* e das marcas presentes nas meta-redes dependentes.

Em termos gerais, a sincronização da evolução dos modelos contidos na superpágina e na meta-rede é realizada em dois instantes especiais, coincidindo o primeiro com a criação da metamarca e activação da meta-rede, por um lado, e o segundo com a remoção da metamarca e desactivação da meta-rede, por outro.

Em secção posterior, serão apresentados os procedimentos de tradução desta forma de especificação hierárquica em termos do modelo básico das RdP-R apresentado em capítulo anterior.

Realça-se a ligação directa entre a existência de uma metamarca e a actividade na meta-rede associada, isto é, considera-se que o metalugar é monomarcado (seguro, para o caso de RdP de baixo-nível).

Importa, pois, analisar o caso em que o lugar possa conter mais do que uma marca (isto é, que seja *k*-limitado, com *k* finito, para o caso de RdP de baixo-nível).

A semântica proposta para esta situação prevê a criação de uma nova instância da meta-rede para cada nova metamarca criada no metalugar. Considerando que o número de marcas no metalugar é finito e conhecido a priori<sup>28</sup>, a situação referida é modelada do seguinte modo:

- criar um lugar complementar ao metalugar,
- gerar uma marcação adequada no lugar complementar, considerando que o metalugar e lugar complementar, bem como todas as transições de entrada e saída a eles ligados, utilizam uma representação estruturada através de vectores; a dimensão dos vectores a utilizar é igual ao número máximo de marcas possível no metalugar inicial.

A Figura 4-29, em que  $N$  representa o número máximo de marcas possíveis em  $p1$ , ilustra o modelo que permitirá modelar a semântica referida. A tradução do modelo da Figura 4-29(a) no da Figura 4-29(b) pode ser embebida nos procedimentos do editor gráfico, tornando o processo transparente para o utilizador.

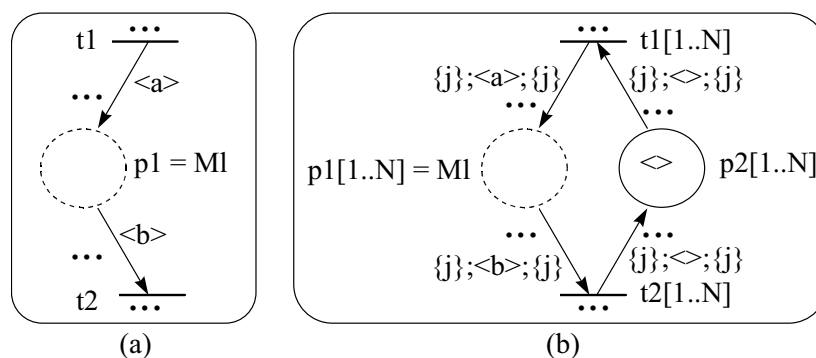


Figura 4-29 - Utilização de metalugares em:

a) lugares monomarcados; b) lugares plurimarcados.

Procedimento alternativo equivalente pode utilizar um modelo “puramente” colorido (sem estruturação) comportamentalmente equivalente ao apresentado. No entanto, nessa solução torna-se necessário alterar a cor associada ao metalugar e as expressões dos arcos a ele interligados, bem como a propagação destas alterações de cor para dentro da meta-rede. As Figura 4-26(b) e Figura 4-26(c) atrás apresentadas ilustram os modelos utilizáveis.

Por questões de sistematização da exposição, adia-se para subsecção posterior a caracterização das meta-redes.

<sup>28</sup> Através da construção do espaço de estados associado à rede da superpágina, por exemplo.

### 4.3.3.b Utilizando metatransições

Uma metatransição permite modelar a evocação síncrona de um metalugar, no sentido em que a activação da meta-rede resulta da remoção simultânea de um conjunto de marcas na superpágina por efeito do disparo de uma transição. Como exemplo de referência, considere-se a evocação de um subprograma com passagem de vários parâmetros de entrada; a modelação dos parâmetros é realizada através das marcas presentes nos lugares de entrada. Quando da evocação, os parâmetros são retirados dos lugares de entrada e dá-se início ao seu processamento, modelado através de um metalugar.

O modelo equivalente a uma metatransição, ao nível da superpágina, é apresentado na Figura 4-30. Para evitar ambiguidades na notação gráfica utilizada e de forma semelhante à utilizada na representação de transições de entrada e saída em macronós, cada arco ligado à metatransição dispõe de uma referência que permitirá identificar qual a transição a que se interliga, bem como a ordem dentro dessa transição. A Figura 4-30 ilustra uma notação possível.

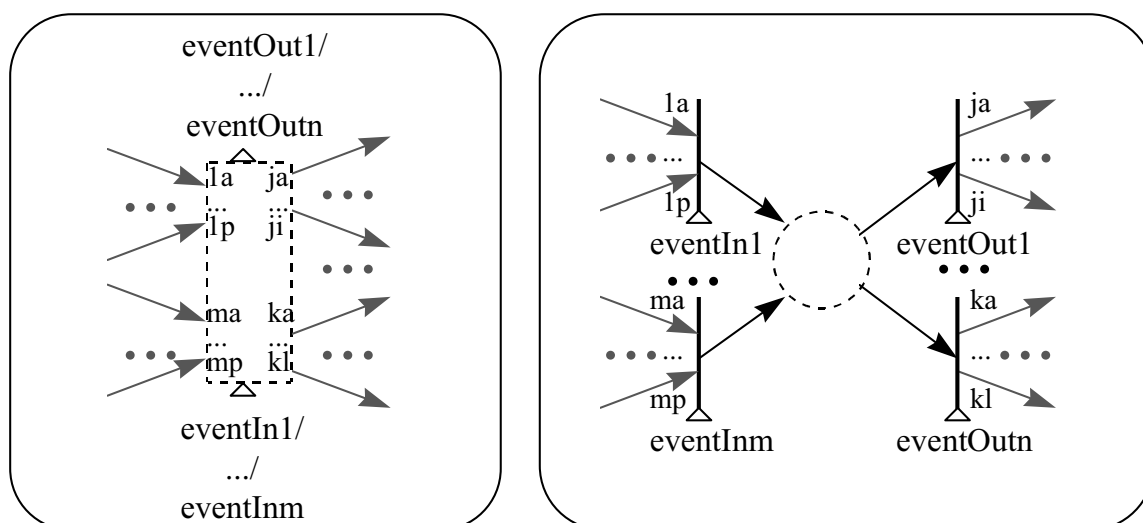


Figura 4-30 - Metatransição: a) representação na superpágina; b) modelo equivalente.

A exemplo do que acontece com os metalugares, também aqui a sincronização da evolução dos modelos contidos na superpágina e na meta-rede associada é realizada em dois instantes especiais, coincidindo o primeiro com a remoção das marcas dos lugares de entrada, criação da metamarca e activação da meta-rede, por um lado, e o segundo com a remoção da metamarca, desactivação da meta-rede e criação das marcas nos lugares de saída, por outro.

Aplicam-se integralmente as considerações produzidas na subsecção anterior relativas à modelação de metalugares marcados com mais do que uma marca, que no contexto presente, permitem modelar a evocação múltipla e simultânea de um mesmo subprograma. Também aqui os procedimentos sintetizados na Figura 4-29 se aplicam.

No sentido de ilustrar a aplicabilidade deste mecanismo de decomposição para além das atitudes de modelação hierárquica, refira-se o interesse, identificado por Carl Adam Petri, em utilizar transições especialmente vocacionadas para a monitoração do estado do sistema, denominadas por transições-mortas (“*dead-transitions*”); resumidamente, são transições que, em situações normais, nunca serão habilitadas, pelo que o seu eventual disparo permite detectar um mau-funcionamento do sistema. A utilização de arcos normais ou de teste, no sentido definido em [Christensen & Hansen, 93], identificando a situação anómala em conjunção com a utilização de uma metatransição permitirá actuar procedimentos de alarme, modelados na meta-rede associada.

Novamente, por questões de sistematização da exposição, a apresentação das características associadas às meta-redes será realizada na subsecção seguinte, que as caracterizará.

#### 4.3.3.c Sobre as meta-redes

Como referido, quer os metalugares, quer as metatransições, isto é, os metanós, têm associados meta-redes, que são subpáginas autónomas. De modo informal, uma meta-rede pode ser caracterizada como uma página comum que contem um conjunto de nós e arcos especiais, responsáveis pela sua sincronização com o metanó associado da superpágina. Assim, a evolução da meta-rede pode ser motivada por uma de duas razões:

- evolução síncrona com o metanó associado;
- evolução autónoma em relação ao metanó associado.

No que se segue, caracteriza-se unicamente o primeiro motivo apresentado, dado o segundo não necessitar de esclarecimentos adicionais (decorre da normal execução da rede).

A evolução síncrona da meta-rede com a superpágina onde o metanó se insere é realizada através do disparo síncrono de transições nos dois níveis, com um de dois objectivos, a saber:

- activação da meta-rede, produzindo uma marcação na meta-rede e a criação de uma metamarca na superpágina;
- desactivação da meta-rede, por força da remoção de metamarca no metanó associado.

As subsecções seguintes iniciam a caracterização das meta-redes associadas aos metanós simples, denominadas por meta-redes simples. Posteriormente, abordar-se-ão as questões referentes às meta-redes associadas aos metanós com história, denominadas por meta-redes com história.

#### 4.3.3.c.1 Inicialização de uma meta-rede simples

A produção de uma marcação inicial numa meta-rede simples é consequência do disparo de uma transição especial de entrada, denominada síncrona de entrada. Na Figura 4-28, devido à simplicidade da situação modelada, a transição síncrona de entrada foi representada por um simples círculo negro (herdado dos statecharts). As características distintivas destas transições são as seguintes:

- não dispõem de arcos de entrada explicitamente especificados pelo projectista (isto é, são transições-fonte);
- o seu disparo é síncrono com o disparo de uma transição na superpágina (modelado através da sua sensibilização em relação a um evento que é emitido quando do disparo da transição da superpágina);
- a lista de características coloridas da marca que chega ao metalugar na superpágina é acessível através de uma lista de variáveis, referidas num atributo da transição que utiliza o identificador “Entrance”;
- o disparo da transição depende da avaliação da guarda associada (como numa transição colorida comum);
- as expressões associadas aos arcos de saída utilizam referências às variáveis referidas no atributo “Entrance” (isto é, a marca referida em “Entrance” comporta-se como o vínculo actual da transição);
- utiliza-se uma notação gráfica distinta para representar este tipo de transição síncrona, constituída por um rectângulo com uma linha longitudinal de simetria.

A Figura 4-31(a) apresenta a notação gráfica utilizada, de acordo com as características enumeradas. Não se faz nenhuma referência explícita ao evento de sincronização, dado este ser de carácter interno, não sendo sequer conveniente a sua indicação pelo modelador do sistema. A expressão “ $k > 2$ ” na Figura 4-31(a) representa a guarda da transição, enquanto as características da marca referida em “Entrance” serão instanciadas aquando da activação da meta-rede e utilizadas como o vínculo actual da transição.

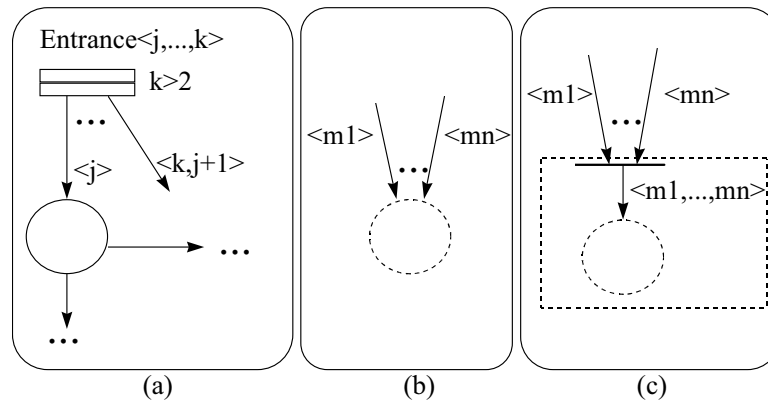


Figura 4-31 - Ativação de uma meta-rede: a) transição de entrada; b) utilização com metalugar; c) utilização com metatransição.

Os atributos de marca colorida referidos em “Entrance” (que são utilizados para gerar as características das marcas a criar pelos arcos de saída da transição síncrona) são determinados:

- no caso de um metalugar (ver Figura 4-31(b)), pela marca transportada pelo arco de chegada ao metalugar na superpágina, quando do disparo da transição associada; no exemplo da Figura 4-31, a marca referida por  $\langle j, \dots, k \rangle$  na Figura 4-31(a) será substituída por uma das marcas  $\langle m1 \rangle, \dots, \langle mn \rangle$  da Figura 4-31(b), de acordo com a transição que ocorra na superpágina;
- no caso de uma metatransição (ver Figura 4-31(c)), pela marca composta pelas características de todas as marcas transportadas pelos arcos de chegada à metatransição, na superpágina; no exemplo da Figura 4-31, a marca referida por  $\langle j, \dots, k \rangle$  na Figura 4-31(a) será substituída pela marca  $\langle m1, \dots, mn \rangle$  da Figura 4-31(c).

Do ponto de vista formal, tudo se passa como se os atributos de cor referidos em “Entrance” fossem parâmetros transportados pelo evento (implícito à execução do modelo) que provoca o seu disparo.

No sentido da máxima simplificação e expressividade gráficas, nos casos em que as características da marcação inicial não dependem das características das marcas que chegam ao metanó, é utilizada uma notação simplificada, semelhante à utilizada nos statecharts, substituindo a transição especial, o atributo *Entrance* e a guarda (sem sentido neste caso) por um simples círculo preto (como utilizado nas Figura 4-28(b) e Figura 4-28(c) atrás apresentadas).

A utilização conveniente da guarda associada à transição especial de entrada, permitirá identificar situações específicas verificadas na superpágina, constituindo-se a meta-rede, nessas situações, num procedimento de utilização excepcional (procedimento de alarme, por exemplo). No caso limite, é possível que, devido à utilização da guarda, não se produza uma marcação inicial na meta-rede,

embora a execução do modelo da superpágina emita o evento que provoca a ocorrência da transição síncrona de entrada; logo, que a meta-rede não seja activada.

Considera-se que as guardas associadas às transições síncronas de entrada utilizadas garantem que o disparo destas transições é mutuamente exclusivo, só podendo, pois, disparar simultaneamente uma ou zero transições síncronas de entrada.

Importa referir dois casos particulares:

- o primeiro, em que um metalugar se encontra inicialmente marcado. Neste caso, a meta-rede encontra-se inicialmente activada, sendo a marcação da meta-rede a correspondente ao disparo da transição síncrona de entrada de acordo com a marcação inicial do metalugar;
- o segundo, em que a meta-rede é, no todo ou em parte, marcada inicialmente com um conjunto especificado de marcas. Neste caso, as transições síncronas de entrada necessitarão de gerar essa marcação inicial nos vários lugares envolvidos. Para isso, basta considerar arcos de saída das transições síncronas de entrada (responsáveis pela activação inicial da sub-rede) com inscrições adequadas à geração da marcação inicial estática especificada; os arcos devem ser replicados para todas as transições síncronas de entrada responsáveis pela activação inicial da sub-rede.

#### **4.3.3.c.2 Desactivação de uma meta-rede simples**

Na presente subsecção apresentam-se os mecanismos associados ao segundo instante de sincronização entre a evolução da meta-rede e da superpágina onde é referida, aquando da desactivação da meta-rede, por força da remoção da metamarca no metanó associado.

Do ponto de vista dos modelos equivalentes dos metanós, aplicáveis na superpágina e tomando como referência a Figura 4-27 e a Figura 4-30, tem-se:

- no caso de um metalugar, este é desmarcado como um lugar comum;
- no caso de uma metatransição, as expressões presentes nos arcos internos, interligando as transições de saída ao metalugar, são todas iguais entre si e semelhantes à apresentada na Figura 4-31(c) para um arco de entrada, referindo as características das marcas utilizadas para o vínculo da transição de entrada.

De modo semelhante ao apresentado em relação ao outro instante de sincronização entre a meta-rede e a superpágina, também este mecanismo de sincronização se baseia na utilização de transições especiais, denominadas síncronas de saída. As características distintivas destas transições são as seguintes:

- o seu disparo é síncrono com o disparo de uma transição na superpágina (por força da sua sensibilização em relação a um evento que é emitido pela transição da superpágina);
- utiliza-se a notação gráfica distinta já utilizada na subsecção anterior para representar este tipo de transição síncrona, constituída por um rectângulo com uma linha longitudinal de simetria;
- possuem um atributo especial, que pode tomar os valores “*Stop*”, “*Reset*” ou “*Exit*”;
- possuem um evento associado, comum a toda a meta-rede, gerado pela superpágina, que, a exemplo das transições síncronas de entrada, está implícito à execução do modelo, não sendo representado pelo projectista.

#### 4.3.3.c.3 Nós da meta-rede

Desta forma, as transições da meta-rede dividem-se nos seguintes tipos:

- simples;
- síncronas de entrada (com o atributo “*Entrance*”);
- síncronas de saída (com o atributo “*Stop*”, “*Reset*” ou “*Exit*”).

A utilização de transições síncronas de saída com os atributos “*Reset*” ou “*Exit*” será realizada posteriormente, sendo o primeiro atributo utilizado para suporte ao conceito de história e o segundo para modelar a conclusão de subprograma. As transições síncronas de saída com o atributo “*Stop*” permitem modelar a desactivação de uma meta-rede simples.

De forma semelhante, os lugares da meta-rede dividem-se em:

- simples;
- de história;
- de controlo.

Quer os lugares de história quer os de controlo não são explicitamente indicados pelo projectista e serão definidos em subsecção posterior (os lugares de história destinam-se a suportar o conceito de história, enquanto os lugares de controlo pertencem à sub-rede responsável pela identificação do estado de actividade da meta-rede).

Para os lugares simples da meta-rede pretende-se que, quando se procede à remoção da metamarca no metanó associado, esses lugares sejam desmarcados. Este comportamento é modelado através da utilização de transições síncronas de saída associadas a todos os lugares simples da meta-rede, cuja expressão de arco, sendo geral, permitirá a remoção de qualquer marca presente no lugar. O evento emitido pela superpágina, com função de desactivação da meta-rede, provocará a desmarcação de todos os lugares.

No caso de utilização de um metalugar, esse evento é gerado aquando do disparo de qualquer transição de saída do metalugar. No caso de uma metatransição, esse evento é gerado aquando do disparo de uma transição de saída.

#### **4.3.3.c.4 Comunicação ascendente**

Como referido, utilizando as técnicas de decomposição vertical é possível proceder à caracterização do sistema a um determinado nível pré-definido. A relação entre a superpágina e as meta-redes associadas é hierárquica, sendo a superpágina autónoma em relação às meta-redes e responsável pela emissão dos eventos que determinam a activação e desactivação das meta-redes. Quando da activação da meta-rede é possível transferir características da superpágina para a meta-rede a activar.

No sentido de permitir um conhecimento do estado da meta-rede ao nível da superpágina, definem-se, também, alguns mecanismos que permitem transferir informação da meta-rede para a superpágina.

A exemplo do que acontece com a situação de activação, também os mecanismos disponíveis para comunicação ascendente estão baseados na utilização da componente não-autónoma do modelo, em particular, de um conjunto de saídas que, sendo activadas pela meta-rede, são sentidas ao nível da superpágina.

São propostos dois mecanismos de notificação da superpágina por parte da meta-rede:

- através da saída booleana *exit*, associada à situação de conclusão normal do subprograma;
- através de um vector de saídas booleanas, genericamente referidas por *Exit[n]*, associadas a situações específicas de marcação da meta-rede.

As semânticas associadas aos dois mecanismos são distintas e detalham-se nos parágrafos seguintes.

O primeiro mecanismo, aplicável nas situações de conclusão da actividade associada à meta-rede, baseia-se na identificação de um conjunto de lugares, designados por lugares de saída, através do atributo *Exit*; acoplado a este identificador pode, eventualmente, encontrar-se a caracterização da cor do lugar. Os lugares de saída são lugares seguros, no sentido que nunca mantêm mais do que uma marca. A saída booleana *exit* é activada quando todos os lugares de saída estiverem marcados. Associado a esta saída está uma estrutura de dados contendo as características das marcas presentes em todos os lugares de saída. Tudo se passa como se existisse uma transição síncrona de saída com atributo “Exit” dispondo de arcos de entrada de todos os lugares de saída; o vínculo associado à situação de transição habilitada é disponibilizado na estrutura de dados associada à saída booleana *exit*. Desta forma, ao nível da superpágina é possível saber do estado de conclusão das actividades, bem como dos resultados alcançados. Ao nível da superpágina a referência a esta saída é realizada através de *exit(marcação)*, permitindo a sua utilização como evento associado às transições de saída do metanó; em “*marcação*” são utilizadas as caracterizações das cores dos lugares (indicadas com os atributos Exit em cada um dos lugares de saída ou nos arcos da transição síncrona de saída, como apresentado na Figura 4-32).

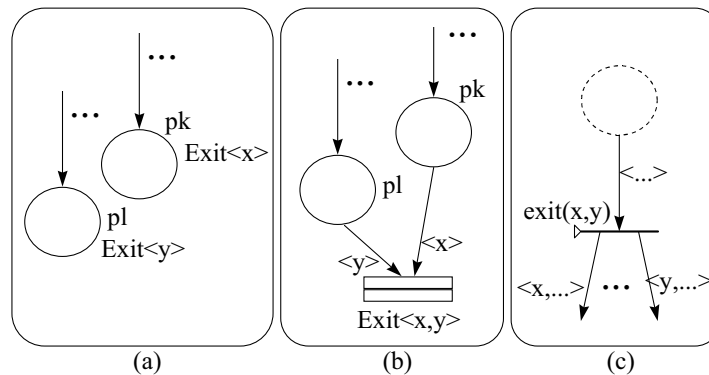


Figura 4-32 - Lugares de saída: a) identificação na meta-rede;  
b) identificação alternativa; c) superpágina.

Como exemplo, ilustrado na Figura 4-32(a), considere-se uma meta-rede com dois lugares de saída, o primeiro contendo o atributo *Exit<x>* e o segundo o atributo *Exit<y>*, em que *<x>* e *<y>* representam as estruturas das cores dos lugares assinalados; o evento *exit(x,y)* disponível na superpágina (ver Figura 4-32(c)), permite concluir sobre a presença simultânea de marcas nos dois lugares, bem como sobre as características da sua marcação actual, através da instanciação das variáveis *x* e *y*. Alternativamente, pode proceder-se à representação explícita da transição síncrona de saída, identificando os lugares de saída, situação que simplifica a notação associada aos lugares de saída, como ilustrado na Figura 4-32(b).

Uma das aplicações deste primeiro mecanismo é o de suportar a implementação do retorno de parâmetros de subprogramas para o programa chamador.

O segundo mecanismo de comunicação ascendente previsto, aplicável na identificação de situações específicas de marcação da meta-rede, baseia-se na activação das saídas *Exit[n]*. Ao contrário do mecanismo de activação tipo “e” da saída *exit*, o mecanismo de activação das saídas *Exit[n]* é do tipo “ou”, sendo activadas quando qualquer dos lugares identificados pelo atributo *Exit(n)* estiver marcado. Às saídas *Exit[n]* não estão associados quaisquer outras características, sendo a sua referência ao nível da superpágina realizada através do evento *Exit[n]* associado às transições de saída do metanó. A saída *Exit[0]* está implicitamente associada ao conjunto de lugares de saída, isto é, identificados com o atributo *Exit*.

Para além dos dois mecanismos referidos, intrínsecos ao modelo considerado, dado a comunicação entre as superpáginas e as meta-redes ser suportada por sinais associados à parte não-autónoma do modelo, outros tipos de dependência podem, facilmente, ser definidos em relação a sinais gerados pela meta-rede, utilizando as características das RdP-R.

#### **4.3.3.c.5 Execução de modelos contendo metanós**

A implementação de especificações contendo metanós é realizada através da execução concorrente dos modelos da superpágina e da meta-rede.

A Figura 4-33 ilustra, simplificada, os mecanismos específicos de sincronização entre as componentes em execução paralela; a informação transportada pelos eventos *Entrance* e *exit*, referida por (m) na figura, representa a transferência de parâmetros entre a superpágina e as meta-redes. Os eventos gerados pela superpágina, identificados com “*Entrance*”, “*Stop*” e “*Reset*”, estão associados às transições síncronas de entrada e saída com atributo do mesmo nome. Caso utilizada, a transição síncrona de saída com o atributo “*Exit*” tem o evento “*Stop*” associado.

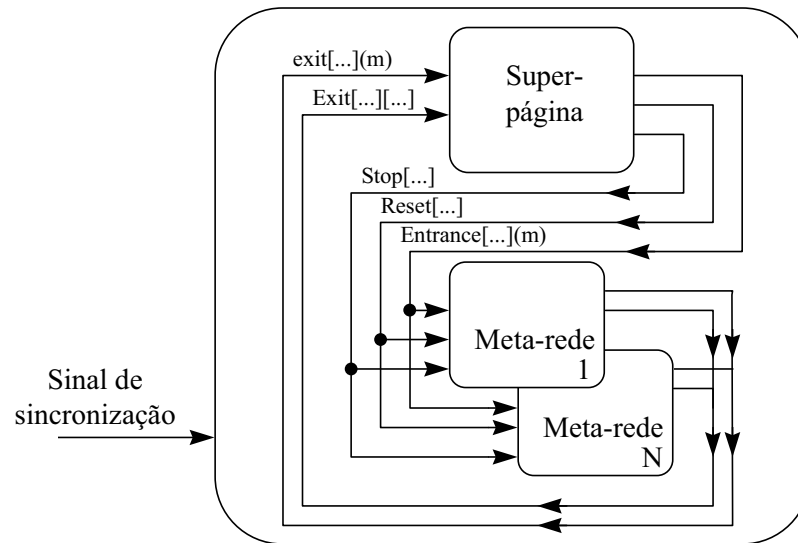


Figura 4-33 - Execução paralela de modelos com metanós.

Importa referir que o número de meta-redes utilizado na execução de uma determinada especificação com metanós depende das marcações verificadas na superpágina, nomeadamente no número simultâneo de metamarcas presentes em cada um dos metanós. Esse número é determinado em tempo de análise da especificação, permitindo calcular a priori os recursos necessários à execução do modelo especificado (recorda-se que esta característica foi identificada como necessária para os sistemas de tempo-real).

Importa, ainda, realçar a capacidade de utilização de referências recursivas, isto é, utilização de referências a metanós com meta-redes associadas coincidentes com a superpágina onde o metanó é referido ou com superpáginas em relação a esta. Obviamente que, para que o número de meta-redes a utilizar na execução seja finito, é necessário que, devido às novas características de marcação verificadas nas meta-redes de níveis mais profundos, as condições para a marcação de um metanó e correspondente criação de uma nova instância da meta-rede se deixem de verificar num nível (finito) da estrutura de dependência hierárquica (situação normal numa função recursiva). Refiram-se duas situações típicas em que essa condição se verifica:

- devido à capacidade de interpretação das características herdadas da superpágina pela meta-rede poderem provocar marcações iniciais distintas na meta-rede, a possibilidade de marcação de um determinado metanó em meta-redes hierarquicamente dependentes pode não ocorrer (cessando o processo de criação de novas meta-redes no processo de execução da especificação);
- como caso extremo da situação anterior, refira-se a possibilidade de nenhuma transição de entrada da meta-rede disparar, por força das guardas utilizadas, garantindo a paragem da criação de meta-redes dependentes.

Na execução do modelo da superpágina devem ser considerados os modelos comportamentais dos metanós referidos na Figura 4-27 e na Figura 4-30.

Na execução da meta-rede deve ser considerado um modelo construído do seguinte modo (apenas parte é explicitamente indicada pelo projectista):

- toma-se, como ponto de partida, o modelo da meta-rede especificado directamente pelo projectista;
- criam-se e interligam-se-lhe três conjuntos de nós e arcos: o primeiro contendo informação sobre o estado de actividade da meta-rede, o segundo responsável pela marcação da meta-rede especificada e o terceiro responsável pela sua desmarcação;
- a sub-rede responsável pela identificação do estado de actividade da meta-rede é constituída por um lugar, com uma interpretação de “meta-rede não-activa” e por um ciclo utilizando uma transição que permite garantir a existência de o máximo de uma única marca nesse lugar; o lugar é marcado inicialmente com uma marca (de baixo-nível);
- a sub-rede responsável pela marcação inicial da meta-rede especificada é constituída pelas transições síncronas de entrada, que recebem arcos do lugar “meta-rede não activa”; o atributo *Entrance* (e seus parâmetros) associado a essas transições é considerado como o evento associado, emitido pela superpágina, para efeitos de disparo das transições; a marca presente no lugar “meta-rede não activa” permite o disparo da transição síncrona de entrada que esteja habilitada pelas condicionantes impostas pela superpágina (como referido, só uma ou zero transições síncronas de entrada ocorrem simultaneamente); a marcação inicial indicada na especificação, se existir, deve ser gerada pelo disparo das transições síncronas de entrada;
- a sub-rede responsável pela desmarcação da meta-rede especificada é constituída por transições síncronas de saída, às quais se associa o evento *Stop*, gerado pela superpágina, ligadas a cada um dos lugares simples, garantindo a sua desmarcação quando da ocorrência do evento *Stop*; caso a transição síncrona de saída com o atributo “*Exit*” seja utilizada considera-se como tendo prioridade inferior em relação às transições síncronas de saída com o atributo “*Stop*” (quando em situação de conflito); as transições síncronas de saída referidas dispõem de arcos de saída para o lugar “meta-rede não activa”.

As transições síncronas referidas são prioritárias em relação às outras transições presentes na meta-rede (de modo a garantir o sincronismo com a superpágina).

Nos procedimentos referidos é pressuposto que a situação de meta-rede completamente desmarcada nunca pode ser atingida. Esta situação deverá ser verificado, através da análise do espaço de estados associado.

Um comentário final sobre a construção do espaço de estados associado à meta-rede, apenas para referir que, desde que se possa garantir o conhecimento prévio do número de valores possíveis que os parâmetros associados ao evento síncrono possam transportar (determinado através da construção do espaço de estados da superpágina), são completamente válidos os procedimentos expostos na subsecção 2.2.11 “Interdependência entre as partes autónoma e não-autónoma”, pg. 57, permitindo “descompactar” as transições síncronas de entrada em transições simples, e posterior utilização do algoritmo de construção do espaço de estados, apresentado na subsecção 3.3.1 “Das RdP-R a máquinas de estados finitos”, pg. 81.

Os procedimentos de obtenção do modelo executável da meta-rede, referidos nos parágrafos anteriores, podem ser sintetizados no modelo da Figura 4-34, aplicado ao exemplo da meta-rede apresentada na Figura 4-28(b).

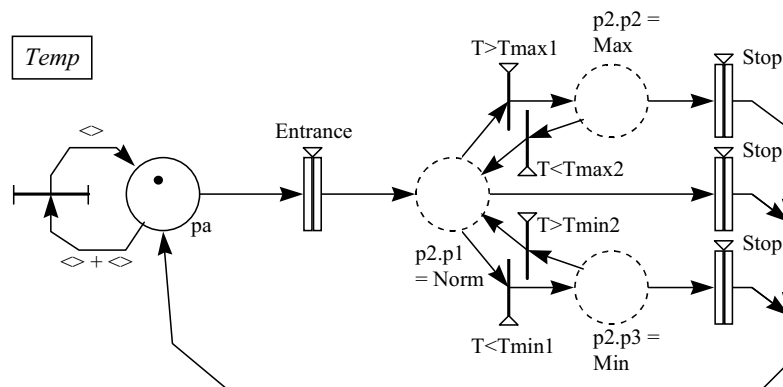


Figura 4-34 - Modelo executável associado à meta-rede *Temp* da Figura 4-28(b).

#### 4.3.3.c.6 Conceito de história

Uma característica comum aos sistemas operativos e linguagens actuais é a capacidade de tratamento de acontecimentos de excepção, através da interrupção do processamento por uma rotina de atendimento do evento.

Em sistemas com condicionantes de tempo-real, como os que se pretendem modelar, a utilização de interrupções e características específicas associadas às rotinas de atendimento ou embebidas no sistema operativo permitem garantir o desempenho adequado do sistema, nomeadamente em termos dos tempos de resposta especificados.

De uma forma geral é, portanto, necessário dispôr da capacidade de interromper uma actividade e retornar posteriormente ao ponto de interrupção.

A solução que se defende neste trabalho é inspirada no mecanismo utilizado nos statecharts e concretizado na definição do conceito de história.

Informalmente, permite que uma meta-rede que tenha sido desactivada por força da ocorrência de um evento que desmarcasse o metanó associado, quando seja novamente activada possa ser inicializada com o estado de marcação que detinha quando da desactivação. É, portanto, um mecanismo que permite associar memória em relação a estados anteriores da meta-rede.

A notação utilizada é semelhante à utilizada nos statecharts, isto é, o metanó, ao nível da superpágina, dispõe de um atributo especial de história, denominado *History*, com a seguinte semântica:

- quando o metanó é assinalado com o valor H para o atributo de história, as activações posteriores da meta-rede associada utilizarão como marcações iniciais as marcações que se verificavam quando da última desactivação da meta-rede;
- quando o metanó é assinalado com o valor H\* para o atributo de história, a propriedade referida no parágrafo anterior aplica-se a todos os metanós referidos na meta-rede, assim como a todos os que venham a ser utilizados nas meta-redes dependentes; consiste na aplicação iterativa do conceito de história a toda a hierarquia dependente do metanó assinalado com o atributo H\*.

De notar que a utilização do mecanismo de *história* deve ser realizada com cuidados especiais do ponto de vista da especificação produzida pelo projectista, garantindo alguns pressupostos para que faça sentido reter informação sobre estados anteriores da meta-rede. De um modo geral, será a aplicação a um problema específico de modelação que determinará a adequação da utilização do conceito de história.

Considerem-se os seguintes exemplos.

Uma situação de interesse é a de modelação de uma actividade através de uma RdP de baixo-nível, que se pretende suspender temporariamente para atendimento de um evento especial, sendo retomada posteriormente no ponto em que tinha sido interrompida. A actividade é modelada através de um metalugar (ou metatransição) e meta-rede associada; ao nível da superpágina utiliza-se uma metamarca de baixo-nível, o que torna as várias execuções da actividade indistintas entre si.

Uma situação distinta é a de evocação de um subprograma com um parâmetro de entrada. Neste caso, a metamarca será colorida e armazenará o valor do parâmetro de entrada. Pode ser, neste caso,

importante garantir que a suspensão temporária da execução da meta-rede activada através da passagem do parâmetro '1', por exemplo, não seja retomada devido à chegada de uma metamarca com o valor '2' e que se utilize uma segunda instância da meta-rede inicializada com o parâmetro '2'. A retoma da primeira instância da meta-rede só será adequada quando da chegada de outra marca '1' ao metalugar. Neste tipo de situação, é necessário garantir que a activação de uma *determinada* instância da meta-rede com história é sempre realizada pela criação de uma metamarca com *determinadas* características.

Uma solução aplicável nestas situações, com características adequadas a ser embebida em procedimentos automáticos associados a um ambiente de apoio à especificação, recorre à utilização de um vector de metanós com história, em número igual ao número de metamarcas possíveis no metanó e de um esquema semelhante ao proposto para utilização em metanós plurimarcados, ilustrado anteriormente na Figura 4-29, em que a marcação inicial do vector de lugares complementares ao vector de metanós é coincidente com as marcações possíveis do metanó, garantindo, desta forma, que o índice do metanó utilizado por uma determinada marca é sempre o mesmo para vários disparos com o mesmo vínculo.

A Figura 4-35 ilustra o procedimento proposto, em que as metamarcas possíveis no metalugar são  $\langle 1 \rangle$ ,  $\langle 2 \rangle$  e  $\langle 3 \rangle$ , reflectido na marcação inicial do lugar vectorizado complementar e em que o valor de  $a$  é determinado pelo vínculo permitido pelos arcos de entrada de  $t1$  (não representados na figura).

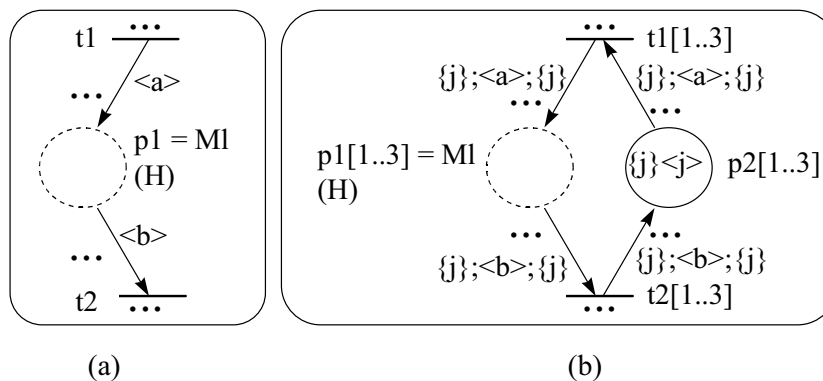


Figura 4-35 - Utilização de metalugares plurimarcados com história:

a) modelo especificado; b) modelo equivalente.

#### 4.3.3.c.7 Utilizando meta-redes com história

Importa demonstrar que a semântica proposta para a utilização do atributo de história pode ser implementada através do modelo RdP-R. Na construção do modelo comportamentalmente equivalente e coerente com a semântica proposta para o conceito de história, devem ser seguidos os seguintes

procedimentos (adicionando e alterando alguns passos aos procedimentos apresentados na subsecção anterior para meta-redes sem atributo de história):

- toma-se, como ponto de partida, o modelo da meta-rede especificado directamente pelo projectista;
- criam-se e interligam-se-lhe quatro conjuntos de nós e arcos: o primeiro permitindo a memorização do estado de marcação quando da última desactivação da meta-rede, o segundo contendo informação sobre o estado de actividade da meta-rede, o terceiro responsável pela marcação inicial da meta-rede especificada e o quarto responsável pela sua desmarcação;
- a sub-rede que permite a memorização do estado de marcação é obtida através da duplicação de todos os lugares simples da meta-rede, criando um conjunto de lugares de história (os metalugares são, para este efeito, considerados lugares simples); os lugares de história herdam, unicamente, as características autónomas dos lugares simples (excluem-se, portanto, a actuação de saídas); criam-se arcos entre os lugares de história e as transições simples, semelhantes aos existentes entre os lugares simples correspondentes e essas transições, nomeadamente em termos das expressões coloridas utilizadas;
- a sub-rede responsável pela identificação do estado de actividade da meta-rede é constituída por dois lugares; o primeiro é marcado inicialmente com uma marca (de baixo-nível) e tem como função modelar o estado de “meta-rede nunca activada”; o segundo tem uma interpretação de “meta-rede não activa” (após a primeira activação);
- a sub-rede responsável pela marcação da meta-rede especificada é constituída por duas sub-redes, a primeira associada à primeira activação da meta-rede e a segunda associada às activações posteriores:
  - a primeira sub-rede é composta pelas transições síncronas de entrada especificadas na meta-rede, que recebem arcos do lugar “meta-rede nunca activada” e geram arcos de saída semelhantes quer para os lugares simples (como indicado na especificação), quer para os lugares de história correspondentes; o atributo *Entrance* (e seus parâmetros) associado a essas transições é considerado como o evento associado, emitido pela superpágina, para efeitos de disparo das transições;
  - a segunda, associada a activações posteriores da meta-rede, é constituída por transições síncronas de entrada, às quais se associa o evento *Entrance* emitido pela superpágina; cada uma dessas transições dispõe de um arco de entrada proveniente

dos lugares de história, dois arcos de saída destinados ao lugar de história e ao lugar simples e um arco proveniente do lugar “meta-rede não activa”, permitindo a marcação dos lugares simples aquando de uma reactivação da meta-rede;

- a sub-rede responsável pela desmarcação da meta-rede especificada é constituída por transições síncronas de saída, às quais se associa o evento *Stop*, gerado pela superpágina, ligadas a cada um dos lugares simples e garantindo a sua desmarcação quando da ocorrência do evento *Stop*; de modo semelhante à situação de meta-rede sem história, as transições síncronas de saída referidas possuem arcos de saída para o lugar “meta-rede não activa”.

As transições síncronas referidas são prioritárias em relação a quaisquer outras transições presentes na meta-rede (de modo a garantir o sincronismo com a superpágina).

O processo referido pode ser ilustrado no modelo da Figura 4-36, aplicado ao exemplo da meta-rede *Temp* apresentada na Figura 4-28(b), caso ao metalugar associado se associe o atributo de história.

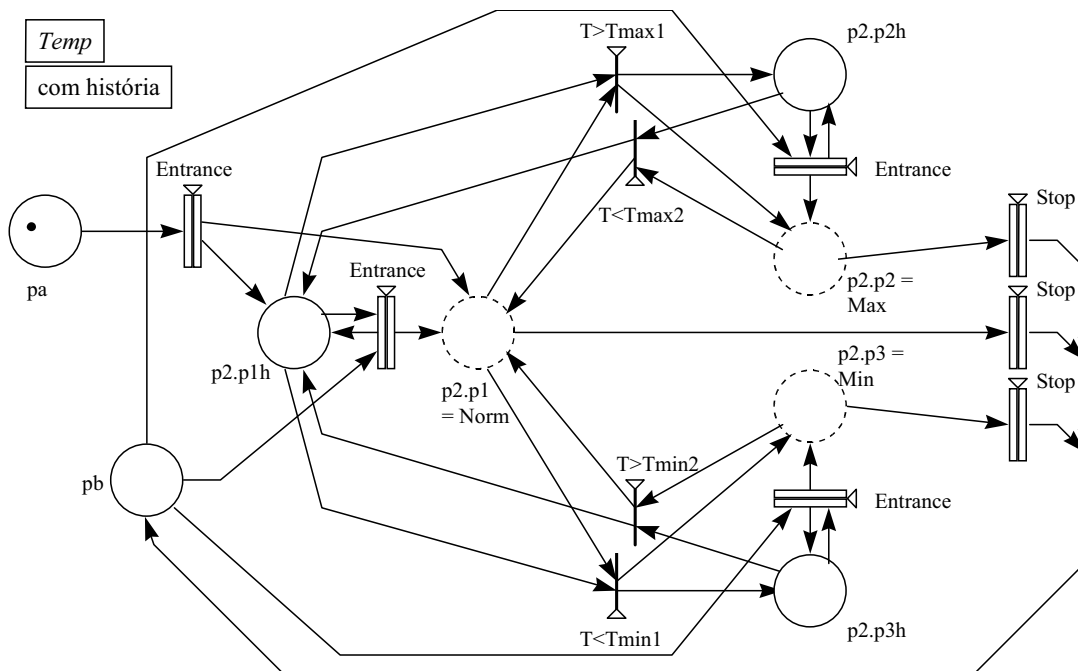


Figura 4-36 - Modelo executável associado à meta-rede *Temp* da Figura 4-28(b)

quando referida com o atributo de história.

#### 4.3.3.c.8 Reiniciando meta-redes com história

Um segundo aspecto a ter em conta prende-se com a capacidade de, tendo definido a utilização de um metanó com história, utilizar, em activações posteriores à primeira, a marcação inicialmente definida

para a situação de “meta-rede nunca activada”. É uma capacidade com interesse para a modelação de controlo, de uma forma geral (como aliás identificado nos trabalhos envolvendo statecharts através da característica de “clear\_history”) e em particular nas situações em que se pretende modelar actividades que, embora interrompíveis, atinjam a situação de conclusão.

A capacidade de reinicialização é, assim, uma característica importante associada ao conceito de história.

Com este objectivo, a superpágina emite o evento “Reset” concomitantemente com o evento “Stop”, forçando, desta forma, não só a desactivação da meta-rede, mas também a sua reinicialização em posterior activação.

Do ponto de vista da superpágina, o evento “Reset” é explicitamente modelado como um evento de saída associado a transições que provoquem a desactivação da meta-rede. Deste modo, quando uma dessas transições ocorre são gerados simultaneamente os eventos “Reset” e “Stop”.

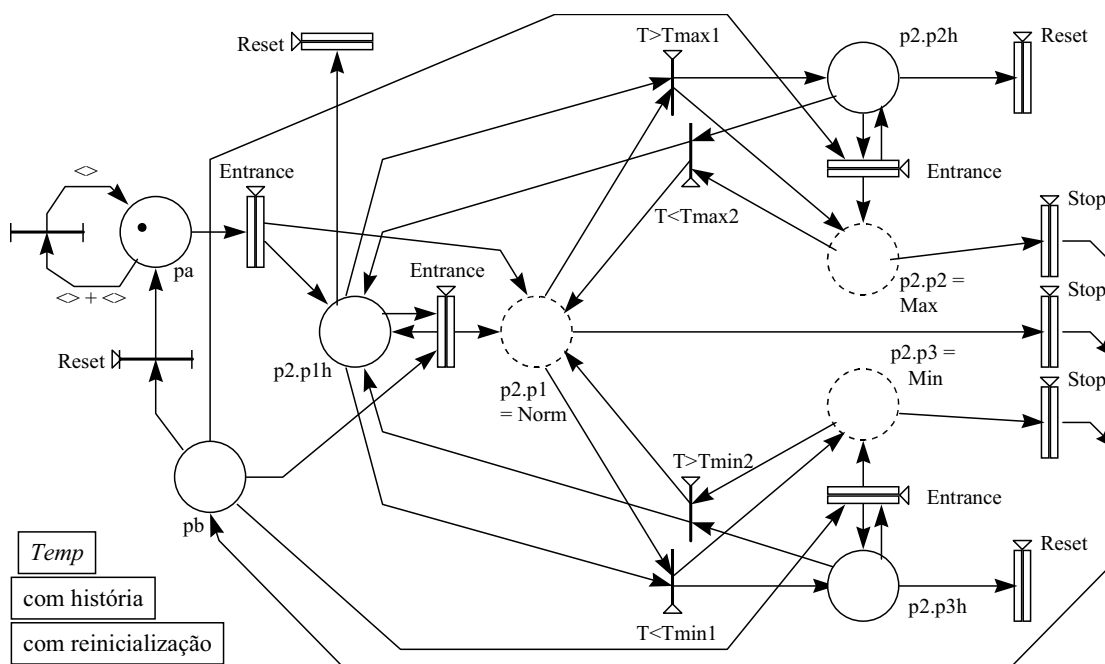


Figura 4-37 - Modelo executável associado à meta-rede *Temp* da Figura 4-28(b) quando utilizada com o atributo de história e capacidade de reinicialização.

Os procedimentos enumerados seguidamente devem ser considerados, em adição aos já apresentados, para a construção do modelo comportamentalmente equivalente suportando a reinicialização de uma meta-rede com história:

- toma-se, como ponto de partida, o modelo da meta-rede comportamentalmente equivalente apresentado na subsecção anterior;

- na sub-rede responsável pela identificação do estado de actividade da meta-rede é adicionada uma transição, à qual se associa o evento “*Reset*”, com um arco proveniente do lugar representando “meta-rede não activa” e um arco de saída para o lugar “meta-rede nunca activada”, permitindo a transferência de todas as marcas entre os dois lugares; de modo semelhante ao modelo utilizado para as meta-redes sem história, é utilizado um ciclo associado ao lugar “meta-rede nunca activada”, com uma transição que permite garantir a existência de o máximo de uma única marca nesse lugar;
- na sub-rede responsável pela desmarcação da meta-rede, adicionam-se transições síncronas de saída, às quais se associa o evento “*Reset*”, gerado pela superpágina, ligadas a cada um dos lugares de história, garantindo a sua desmarcação quando da ocorrência do evento *Reset*;

O processo referido pode ser ilustrado no modelo da Figura 4-37, aplicado ao exemplo já utilizado da meta-rede *Temp* apresentada na Figura 4-28(b), caso ao metalugar associado se associe o atributo de história e se pretenda utilizar a capacidade de reinicialização.

#### 4.3.3.d Utilizando meta-redes solidárias

Como as próprias RdP evidenciam e dão suporte, um requisito comum à especificação de sistemas envolve a modelação de actividades paralelas. Em não raras situações, a activação e desactivação dessas actividades paralelas deve ser realizada simultaneamente; essas actividades são, aqui, denominadas por actividades solidárias. Como exemplo de formalismo de suporte a essas situações, considerem-se os *Grafcet* [Grafcet, 92], formalismo largamente utilizado (predominantemente em França) para a especificação de controladores lógicos programáveis e inspirado nas RdP Ordinárias. Como segundo exemplo significativo, refiram-se os statecharts e a sua característica de ortogonalidade, modelada através de agrupamentos-e.

Sendo a meta-rede um submodelo hierarquicamente dependente da actividade de uma superpágina, cujo estado de execução é activado e desactivado pelo disparo de transições na superpágina, é fácil acomodar a capacidade de modelar actividades solidárias.

Desta forma, implícito à técnica aqui proposta de decomposição vertical, é possível definir um conjunto de meta-redes solidárias e associá-las a um único metanó. Cada uma das meta-redes solidárias é responsável pela modelação de uma actividade solidária.

Todas as meta-redes solidárias possuem um interface uniforme, isto é, as marcas recebidas (e emitidas) pelos metanós associados têm características semelhantes.

É da responsabilidade da superpágina a emissão síncrona dos eventos de activação e desactivação das várias meta-redes solidárias.

Do ponto de vista da notação gráfica utilizada, duas alterações são de referir:

- a primeira ao nível da superpágina, em que a lista de identificadores das meta-redes solidárias associadas deve ser referida junto do identificador do supernó;
- a segunda ao nível da subpágina, em que a representação das várias meta-redes solidárias pode ser realizada numa mesma página, sendo separadas por uma linha a tracejado (a exemplo do utilizado nos statecharts com o mesmo objectivo); alternativamente, as várias meta-redes solidárias podem ser representadas em páginas distintas.

#### *4.3.3.e Análise breve de aplicações*

Nas subsecções seguintes procede-se à apresentação simplificada da aplicação dos mecanismos apresentados de decomposição vertical a várias situações comuns de modelação de sistemas.

##### **4.3.3.e.1 Aplicação à modelação de sistemas de tempo-real**

Serve a presente subsecção para ilustrar a aplicação dos mecanismos apresentados de decomposição vertical à modelação de sistemas de tempo-real.

Como referido em subsecção anterior, as capacidades de interrupção e posterior retoma da actividade de um processo são fundamentais para aplicações a este tipo de sistemas.

Em relação à primeira capacidade, relativa à capacidade de um processo poder interromper outro, duas atitudes básicas podem ser consideradas [Berry, 93]:

- interrupção forçada<sup>29</sup>;
- interrupção permitida<sup>30</sup>.

Esta classificação enfatiza a característica de “quando a interrupção pode ocorrer”. Enquanto a interrupção forçada se apresenta estritamente necessária para a implementação de sistemas reactivos ou de tempo-real, a segunda está presente na maioria dos sistemas operativos actuais. Como exemplos

---

<sup>29</sup> Tradução de “must preemption”.

<sup>30</sup> Tradução de “may preemption”.

da utilização de mecanismos tipo “interrupção permitida”, refiram-se os associados à interacção com o utilizador em sistemas operativos tipo UNIX ou Windows, onde, por exemplo, a execução de uma ordem do utilizador para cancelar a execução de um programa não tem garantias de ser cumprida (pelo menos, de forma imediata).

Uma segunda classificação dos métodos de interrupção, relacionada com a forma como a retoma da execução do processo é ou não executada, enfatiza a característica “como a interrupção pode ocorrer”. Duas atitudes básicas são consideradas:

- interrupção com suspensão (ou suspensão temporária);
- interrupção com cancelamento (ou cancelamento).

Enquanto na primeira atitude, o estado em que se abandona o processo é preservado para utilização quando da retoma (isto é, a actividade é suspensa temporariamente), na interrupção com cancelamento a actividade é suspensa permanentemente (isto é, cancelada).

Os mecanismos propostos de decomposição vertical suportam completamente as quatro atitudes apresentadas, como se ilustrará seguidamente, considerando que o processo que se pretende interromper está modelado através de uma meta-rede:

- o mecanismo de interrupção forçada é modelado através dos eventos gerados na superpágina que provocam a desactivação da meta-rede; a transição  $t_1$  da Figura 4-38 ilustra esta situação;
- o mecanismo de interrupção permitida é modelado através de eventos gerados na superpágina (que modelam o pedido de interrupção), condicionados pela verificação de “estados receptivos à interrupção” na meta-rede; os “estados receptivos à interrupção” são sinalizados através dos sinais de saída fornecidos pela meta-rede; a transição  $t_2$  da Figura 4-38 ilustra esta situação, em que  $Exit[1]$  é uma saída gerada pela meta-rede, quando da marcação de lugares com o atributo  $Exit(1)$ ;
- o mecanismo de suspensão temporária é modelado através da utilização do conceito de história, podendo ser aplicado apenas à meta-rede imediatamente dependente ou a toda a estrutura dependente de um metanó, através da utilização do atributo de história com valor  $H$  ou  $H^*$ , respectivamente;

- o mecanismo de interrupção com cancelamento decorre da utilização simplificada de metanós (isto é, sem recurso ao atributo de história), por força de eventos gerados na superpágina que provocam a desactivação da meta-rede.

Importa referir que a utilização facilitada destes vários mecanismos só poderá ser atingida quando convenientemente embebida num ambiente de desenvolvimento gráfico em que as várias características possam ser facilmente tratadas (no sentido de tratadas quase-transparentemente).

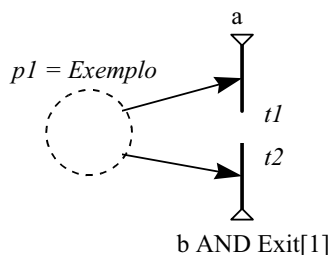


Figura 4-38 - Modelação de interrupções forçada e possível.

#### 4.3.3.e.2 Modelação de subprogramas

A modelação de subprogramas e das capacidades de interrupção associadas é completamente suportada pelos mecanismos de decomposição vertical propostos.

Seguidamente analisam-se brevemente os aspectos considerados mais relevantes e suporte de modelação possíveis:

- a modelação de sub-rotinas é suportada pela utilização de metanós; a criação de metamarcas ao nível da superpágina representa a chamada, enquanto a meta-rede representa a actividade específica da sub-rotina; a destruição da metamarca modela a saída da sub-rotina; a passagem de parâmetros de entrada e saída é modelada através da utilização de marcas que “entram” ou “saem” do metanó; a transferência de parâmetros de entrada para a sub-rotina é realizada através de variáveis da parte não-autónoma do modelo (nomeadamente os parâmetros associados aos eventos síncronos), garantindo que a criação da marca na superpágina seja simultânea com a activação da meta-rede; a transferência de parâmetros de saída da sub-rotina é realizada através de variáveis da parte não-autónoma do modelo que garantem a transferência das características das marcas presentes nos lugares de saída da meta-rede (identificados com o atributo “Exit” ou ligados à transição síncrona de saída com o atributo “Exit”) para os lugares de saída do metanó na superpágina; a notificação da superpágina por parte da meta-rede indicando que o processamento associado à sub-rotina está concluído concretiza-se através da variável de

saída da meta-rede denominada “*exit*”, que para o efeito deverá ser utilizada na superpágina (ver transição  $t2$  da Figura 4-39(a));

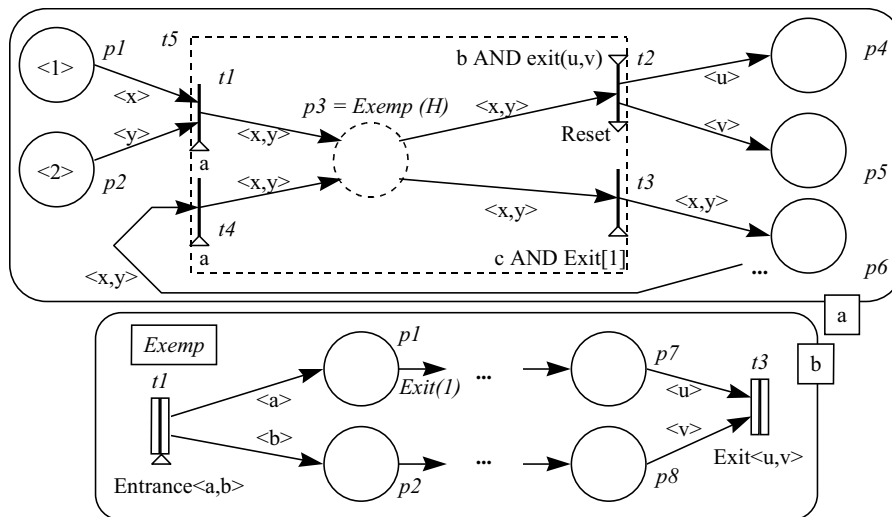


Figura 4-39 - Modelação de chamada de sub-rotina.

- a “interrupção forçada” e a “interrupção permitida” da sub-rotina são, em sintonia com o apresentado na subsecção anterior, modeladas pelas transições de saída do meta-nó; quando se pretenda que a sub-rotina possa ser interrompida apenas em determinados estados do seu funcionamento, os eventos associados às transições de saída devem ser condicionados com a utilização das saídas *Exit*[*n*] geradas pela meta-rede, quando da marcação dos lugares possuidores do atributo *Exit*(*n*) (ver transição  $t3$  da Figura 4-39(a));
- a capacidade de “suspensão temporária”, de novo de acordo com o apresentado na subsecção anterior, é garantida através da utilização do conceito de história associado ao metanó modelando a actividade da sub-rotina; as interrupções da sub-rotina, que não pelo efeito da ocorrência da transição associada a conclusão normal, provocam uma memorização do estado de marcação, que será retomado quando de posteriores activações da sub-rotina, entendidas, neste caso, como retorno ao ponto de interrupção (ver transição  $t4$  da Figura 4-39).

Na Figura 4-39 ilustra-se a chamada de uma sub-rotina modelável através da utilização do metalugar  $p3$  (ou alternativamente, através da metatransição  $t5$ ) com dois parâmetros de entrada e dois de saída.

A chamada da sub-rotina ocorre quando os lugares  $p1$  e  $p2$  estiverem marcados e o evento  $a$  ocorra; no caso ilustrado os valores dos parâmetros de entrada (associados ao evento “Entrance”) são ‘1’ e ‘2’.

A transição  $t_2$  ocorre quando ocorrerem os eventos  $b$  e  $exit$  (gerado pela meta-rede quando os lugares de saída se encontrarem marcados); modela a conclusão normal da sub-rotina com a devolução dos resultados ao programa chamador. De notar que a remoção da metamarca, por conclusão da sub-rotina, gera o evento de *Reset*, responsável pela reinicialização da meta-rede, de modo a esta poder ficar receptiva a novos parâmetros de entrada.

A sequência de ocorrências  $t_3$ ,  $t_s$ ,  $t_4$  (em que  $t_s$  representa qualquer sequência possível de ocorrências na superpágina onde não se inclui a ocorrência da transição  $t_1$ ) permite modelar a interrupção com suspensão temporária da sub-rotina em pontos específicos da sua actividade (através da avaliação de *Exit[1]* que é emitido pela meta-rede) e posterior retoma através de remarcação do metalugar.

#### 4.3.3.e.3 Modelação de statecharts através de RdP-RH

Tendo os trabalhos sobre statecharts sido um dos pontos de partida para as propostas de mecanismos de estruturação do modelo aqui apresentadas, importa verificar o nível de sobreposição entre os statecharts e as RdP Reactivas Hierárquicas, quer ao nível das capacidades de modelação, quer ao nível das possibilidades de tradução mútua.

Sai, obviamente, fora do âmbito deste trabalho identificar uma semântica específica para os statecharts, bem como a enumeração exhaustiva dos procedimentos de modelação das várias características dos statecharts através de RdP-R. Justifica-se, deste modo, que na presente subsecção se abordem apenas as características mais “fortes” dos statecharts.

Com estes pressupostos, os statecharts podem ser vistos como uma sub-classe das RdP-RH. No entanto, a modelação baseada em máquinas de estado revela-se mais adequada para a modelação de fluxos de controlo e pouco adequada para modelar recursos, onde as RdP Lugar-Transição e Coloridas são, comumente utilizadas. É possível traduzir um modelo expresso em statechart para um modelo RdP-RH, pelo que as suas capacidades de modelação são equivalentes (no mesmo sentido em que a capacidade de modelação de uma RdP Colorida é semelhante à de uma RdP Lugar-Transição).

Assim, na presente subsecção proceder-se-á a apresentação de procedimentos de tradução de algumas características de uma especificação utilizando statecharts no modelo RdP-RH. A metodologia aqui seguida para apresentação desses procedimentos baseia-se em [Gomes & Steiger-Garção, 96c] e na análise de situações específicas de modelação, traduzidas num conjunto de regras de tradução.

Regra 1 - um estado do statechart será traduzido num lugar da RdP-RH; um agrupamento-ou ou um agrupamento-e serão traduzidos num metalugar da RdP-RH.

O conjunto seguinte de regras refere-se unicamente a agrupamentos-ou que não utilizem o conceito de história. Posteriormente abordar-se-ão os agrupamentos-e e o suporte ao conceito de história. Consideram-se três grupos de situações típicas:

- no primeiro grupo são considerados os arcos que partem ou chegam a um estado, interligando estados e agrupamentos ao mesmo nível, isto é, internos a um mesmo agrupamento-ou; aplica-se a regra 2;
- no segundo grupo consideram-se os arcos que partem ou chegam ao limite exterior de um agrupamento-ou; correspondem a “ligações bem-estruturadas”, isto é, associadas a um encapsulamento hierárquico dos estados em agrupamentos-ou; a regra 3 aplica-se a arcos que partem do agrupamento-ou enquanto a regra 4 está associada aos arcos que chegam ao agrupamento-ou;
- o terceiro grupo engloba as situações “mal-estruturadas” em que os arcos chegam ou partem de um estado interno do agrupamento-ou; aplicam-se as regras 5 e 6, respectivamente.

Regra 2 - um arco do statechart que chega a (parte de) um estado E, interligando estados ou agrupamentos ao mesmo nível, é traduzido por uma transição com arco de saída (entrada, respectivamente) ligado ao lugar equivalente ao estado E; o evento associado ao arco passa a ser o evento associado à transição.

Regra 3 - um arco do statechart que parta do limite do agrupamento-ou será traduzido por uma transição com arco de entrada ligado ao metalugar equivalente ao agrupamento de partida; o evento associado ao arco passa a ser o evento associado à transição.

Regra 4 - um arco do statechart que chegue ao limite do agrupamento-ou será traduzido por uma transição com um arco de saída para o metalugar equivalente ao agrupamento de chegada; o evento associado ao arco passa a ser o evento associado à transição.

Regra 5 - um arco do statechart que chegue a um estado E (ou agrupamento) interno a um agrupamento-ou A é traduzido por uma transição com arco de saída para o metalugar equivalente ao agrupamento A de chegada; o evento associado ao arco passa a ser o evento associado à transição; a expressão do arco permitirá que, em conjunção com as guardas a utilizar nas transições síncronas de entrada da meta-rede, seja marcado o lugar equivalente ao estado E.

Regra 6 - um arco do statechart que parta de um estado E (ou agrupamento) interno a um agrupamento-ou A é traduzido por uma transição com arco de entrada do metalugar equivalente ao

agrupamento A; considerando que o estado E é o único a possuir o atributo  $Exit(k)$ , o evento associado ao arco passa a ser composto pelo evento inicialmente associado à transição com a saída gerada pela meta-rede  $Exit[k]$ .

O suporte ao conceito de história é conseguido através da transferência dos atributos de história H ou  $H^*$  associados a agrupamentos do statechart para os metalugares equivalentes. A modelação do evento *clear\_history*, permitindo reutilizar a marcação inicial do agrupamento, é conseguida através da capacidade de reinicialização da meta-rede com recurso ao evento *Reset*.

No tocante à modelação de agrupamentos-e, em que se modela a activação e desactivação sincronizada de várias máquinas de estados paralelas, o suporte à sua modelação baseia-se na utilização de meta-redes solidárias. Cada um dos componentes ortogonais do agrupamento-e é traduzido numa meta-rede comportamentalmente equivalente (utilizando as regras apresentadas). Cada meta-rede produzida é considerada solidária. A lista dos identificadores das meta-redes solidárias deve ser referida junto ao metalugar representativo do agrupamento-e.

No tocante à emissão de eventos devido à mudança de estado do statechart, o suporte à sua modelação baseia-se simplesmente em os associar às transições equivalentes que modelam os arcos envolvidos.

O conjunto de procedimentos apresentado permite traduzir a estrutura de um statechart numa estrutura equivalente em RdP-RH. A definição da marcação inicial a utilizar é realizada unicamente de acordo com o estado definido no statechart, no modelo do topo da hierarquia.

A Figura 4-40 apresenta um exemplo simples de um statechart e o modelo RdP-RH equivalente, resultante da aplicação do conjunto de regras de tradução apresentado.

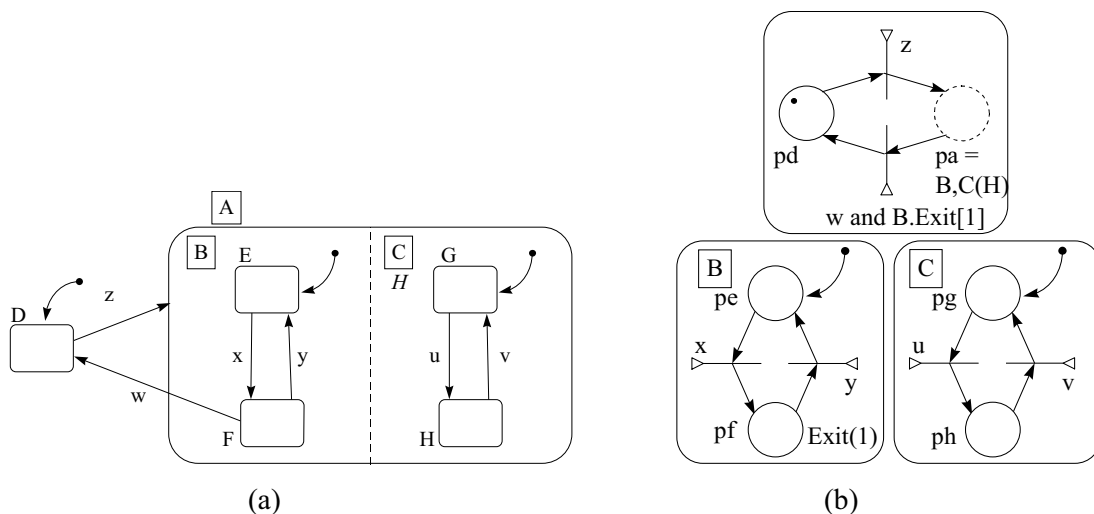


Figura 4-40 - Exemplo de modelos comportamentalmente equivalentes:

a) statechart; b) páginas de RdP-RH.

A Figura 4-41(a) apresenta a representação textual para o statechart da Figura 4-40(a), de acordo com o formato utilizado em [Soares, 97] e considerando que os eventos de entrada são mutuamente exclusivos. Da sua análise resulta o espaço de estados da Figura 4-41(b).

A execução da RdP-RH da Figura 4-40(b) é realizada através da execução paralela das três páginas indicadas (de acordo com a Figura 4-33 atrás apresentada). Como referido em capítulo anterior, a execução das RdP-R pode ser realizada com recurso a um jogador-de-marcas ou a um jogador-de-estados; no caso em análise, as duas soluções coincidem, dado que cada página é uma máquina de estados.

#### **4.3.4 Decomposição mista**

As duas técnicas de decomposição apresentadas, denominadas por horizontal e vertical, utilizam os denominados macronós e metanós, respectivamente. Enquanto na primeira técnica a execução do modelo é realizada ao nível do modelo plano, na segunda técnica procede-se a uma estratificação do modelo e a execução do modelo é realizada através de especificações paralelas estruturadas hierarquicamente, com procedimentos de sincronização específicos, nomeadamente a interrupção de um processo por outro hierarquicamente superior.

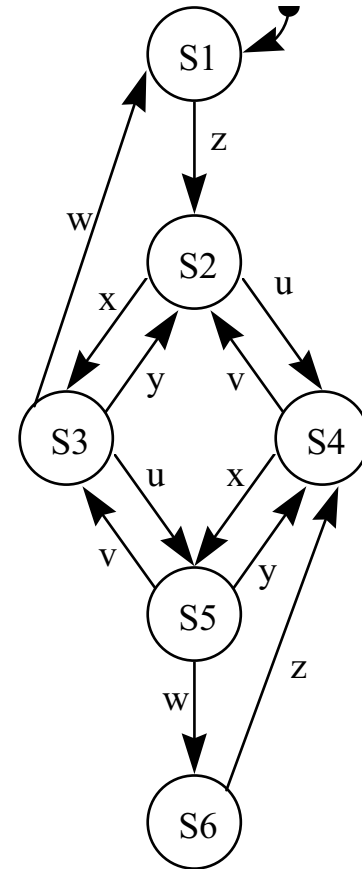
Desta forma, enquanto na primeira técnica a ênfase é colocada na decomposição para efeitos de especificação, podendo ser encarada como um expediente fundamentalmente gráfico, na segunda técnica a ênfase é colocada na decomposição para efeitos de execução do modelo (simulação ou implementação).

```

declare {
  events {
    x is external;
    y is external;
    z is external;
    w is external;
    u is external;
    v is external;
    exclusive x,y,z,w,u,v;
  }
}
cluster universe(d,a) is {
  state d {
    z -> ::universe.a;
  }
  set a(b,c) is {
    cluster b(e,f) is {
      state e {
        x -> f;
      }
      state f {
        y -> e;
        w -> ::universe.d;
      }
    }
    cluster c(g,h) history is {
      state g {
        u -> h;
      }
      state h {
        v -> g;
      }
    }
  }
}

```

(a)



(b)

Figura 4-41 - Statechart da Figura 4-40(a):

a) sua representação textual; b) espaço de estados associado.

Do ponto de vista da caracterização das subpáginas associadas a estes nós, em termos dos nós de interface com a superpágina, encontram-se situações duais. Enquanto no caso das macro-redes utilizadas na decomposição horizontal, os nós-fronteira são lugares, no caso das meta-redes utilizadas na decomposição vertical, os nós-fronteira são transições síncronas.

Como referido, a obtenção do modelo plano para execução das especificações com macronós, faz-se através de um processo de inserção da macro-rede na superpágina onde é referida. A utilização de

uma macro-rede com uma semântica de lugar implica que as instâncias dos lugares-fronteira existam na própria macro-rede, enquanto que caso se pretenda utilizar uma semântica de transição, os lugares-fronteira pertencem à superpágina.

Embora no caso de utilização da decomposição vertical o processo de execução seja distinto e as transições-fronteira sejam nós da sub-rede autónoma que implementa a meta-rede, a utilização de uma semântica de lugar em relação a uma meta-rede deixa a ideia de que as transições-fronteira existem, na superpágina, fora do metalugar; caso se utilize uma semântica de transição em relação a uma meta-rede, as transições-fronteira existem na metatransição.

Do ponto de vista das arquitecturas para execução, a execução de especificações com macronós é realizada através do modelo plano associado, enquanto a execução de especificações com metanós necessita de uma arquitectura permitindo a execução concorrente das várias páginas especificadas hierarquicamente, dispondo de capacidades de sincronização mútua.

A terceira técnica de decomposição proposta, denominada por mista, resulta da integração de características específicas das duas técnicas básicas referidas e pode ser descrita como a utilização concomitante das capacidades de execução de um modelo plano associado a uma especificação hierárquica, tornada disponível pela utilização de macronós, com as capacidades de agrupar nós e especificar de modo compacto evoluções comuns a esse grupo de nós, tornada disponível pela utilização de metanós.

De algum modo, trata-se de disponibilizar mecanismos tipo “interrupção” para serem aplicados a estruturas tipo “macro”. Uma segunda leitura possível pode considerá-la, simplesmente, como uma técnica de implementação de metanós utilizando estratégias de macronós.

Considerou-se o facto das arquitecturas de referência para a execução serem distintas como suficientemente significativo para a sua identificação como uma técnica de direito próprio.

Os nós de suporte a esta forma de decomposição são denominados por supernós (por contraponto aos macronós e aos metanós).

Assim, esta técnica de decomposição pode ser considerada como uma generalização das duas técnicas anteriormente propostas, no sentido em que, no que diz respeito à forma de produzir uma especificação, as duas técnicas propostas podem ser consideradas como casos particulares desta terceira técnica:

- dado a execução ser realizada ao nível do modelo plano, é possível encontrar um modelo utilizando macronós comportamentalmente equivalente a um modelo utilizando supernós;
- dada a utilização de uma estruturação hierárquica, no sentido de condicionar evoluções na subpágina por evoluções verificadas na superpágina, é possível encontrar um modelo utilizando metanós comportamentalmente equivalente a um modelo utilizando supernós.

Já no tocante à execução dos modelos associados, existe uma diferença significativa entre as várias técnicas. Na técnica de decomposição mista, a execução da especificação produzida é realizada ao nível do modelo plano (de modo semelhante ao utilizado para os macronós), pelo que a arquitectura de suporte à sua execução não necessita de características especiais (como a utilizada com os metanós). Esta é a característica distintiva em relação à decomposição vertical, permitindo que seja encarada como uma forma de implementação de metanós com arquitecturas de execução aplicáveis a macronós.

Os supernós podem ser de dois tipos:

- superlugares;
- supertransições.

Ao nível da superpágina, os superlugares interligam-se a nós com semântica de transição, enquanto as supertransições se interligam a nós com semântica de lugar.

Os supernós dispõem de uma notação gráfica distinta, como apresentado na Figura 4-42. As subpáginas associadas a estes supernós são denominadas por super-redes.



Figura 4-42 - Representação de supernós: a) superlugar; b) supertransição.

Importa realçar que, embora esta terceira técnica não adicione capacidades de modelação em relação às duas inicialmente apresentadas, o seu uso pode ser de grande utilidade para o projectista, permitindo implementar uma especificação com metanós através da execução de uma única RdP-R.

Nas subsecções seguintes caracterizam-se os supernós e as super-redes associadas, bem como se ilustra a sua utilização no suporte à modelação de statecharts através de RdP-R.

#### 4.3.4.a Sobre as super-redes

Como já referido, as super-redes, sendo as subpáginas associadas aos supernós, apresentam as características comportamentais das meta-redes apresentadas anteriormente. Desta forma, do ponto de vista da especificação da subpágina por parte do utilizador aplica-se a descrição das meta-redes atrás realizada.

Assim, as transições da super-rede dividem-se em:

- simples;
- síncronas de entrada (com o atributo “*Entrance*”);
- síncronas de saída (com o atributo “*Stop*”, “*Reset*” ou “*Exit*”).

As transições síncronas de entrada e de saída são utilizadas no interface da super-rede e responsáveis pela sua interligação com a superpágina, enquanto as transições simples são internas à super-rede.

De forma semelhante, os lugares da super-rede dividem-se em:

- simples;
- de história;
- de controlo.

Os lugares simples comportam-se como lugares vulgares, com a excepção de serem desmarcados por acção do disparo de uma transição síncrona de saída com o atributo “*Stop*” ou “*Exit*”, como caracterizado quando da apresentação dos metanós.

Os lugares de história comportam-se como a memória dos estados de marcação dos lugares simples, sendo utilizados implicitamente na execução quando se utilizarem supernós com o atributo de história, no sentido proposto quando da apresentação de metanós com história. Não são explicitamente representados pelo projectista, sendo unicamente utilizados na execução da especificação e são desmarcados por acção do disparo de transições síncronas de saída que gerem o evento de “*Reset*”.

Os lugares de controlo são utilizados na modelação do estado da actividade da super-rede e não são especificados explicitamente pelo projectista, sendo utilizados implicitamente quando da execução.

No sentido de evitar repetições, considera-se válido, de uma forma geral, o que foi escrito para a técnica de decomposição vertical atrás proposta, sendo necessário explicitar o que de diferente existe nesta técnica, ou seja, a execução da especificação com base no modelo plano. Dito de outra forma, importa esclarecer os procedimentos de inserção das subpáginas nas superpáginas, que será o assunto da próxima subsecção.

#### *4.3.4.b Procedimentos de inserção das subpáginas*

Optou-se por fazer a análise separada de cada uma das características da subpágina, em termos dos procedimentos que permitem a sua inserção na superpágina.

A inserção da super-rede na superpágina onde é referida é realizada, em termos gerais, de modo semelhante à indicada para a inserção dos macronós nas superpáginas, precedido da obtenção do modelo plano comportamentalmente equivalente à subpágina.

Os modelos equivalentes apresentados para a execução dos metanós são considerados como o ponto de partida para a obtenção do modelo plano, ao que se adicionam os seguintes procedimentos, divididos em três grupos:

- o primeiro aplica-se aos nós internos à super-rede,
- o segundo às transições síncronas de entrada e
- o terceiro às transições síncronas de saída.

Seguidamente analisam-se estes três grupos de procedimentos.

##### **4.3.4.b.1 Modelando os nós internos às sub-redes**

No que se refere aos nós internos (transições simples e lugares), o procedimento é de os considerar no modelo a inserir na superpágina preservando todas as suas características, quer de topologia, quer de atributos.

No tocante aos lugares de controlo, no sentido de possibilitar o teste completo ao estado da sub-rede (e dada a incapacidade de teste de zero marcas), torna-se necessário dispôr de um estado cuja interpretação associada seja “super-rede activa”; esse estado é modelado por um lugar complementar aos lugares “super-rede nunca activada” e “super-rede não activa”, referidos em subsecções anteriores (quando da análise de meta-redes), como os lugares “meta-rede nunca activada” e “meta-rede não activa”. A interligação deste lugar é realizada do modo usual para os lugares complementares, isto é,

considerando arcos de entrada com origem em todas as transições de saída dos referidos lugares e arcos de saída para todas as transições de entrada dos referidos lugares.

#### **4.3.4.b.2 Modelando a activação das sub-redes**

No tocante aos arcos que fazem chegar marcas à fronteira do supernó, isto é, às transições síncronas de entrada da super-rede, o modelo plano comportamentalmente equivalente à semântica proposta para a sua interpretação, baseia-se em considerar um mecanismo de fusão de transições entre as transições da superpágina e as transições síncronas da subpágina.

Considere-se uma transição de entrada  $t_1$  no supernó e um conjunto de  $N$  transições síncronas de entrada. O modelo resultante da inserção da subpágina substitui a transição de entrada  $t_1$  por um vector de transições  $t_1[1..N]$ , em que os arcos de entrada são fornecidos a todas as transições do vector. Cada uma das transições do vector será fundida com uma transição síncrona de entrada, isto é, serão consideradas uma única. Eventos associados à transição  $t_1$  serão associados a todas as transições  $t_1[1..N]$ . As guardas de cada uma das transições do vector resultam da agregação das guardas da transição  $t_1$  e de cada uma das guardas especificadas nas transições síncronas. Os arcos (de entrada e saída) da transição síncrona de entrada são ligados à transição “fundida”. O lugar modelando “super-rede não activa” deve ser considerado como lugar de entrada para todas as transições fundidas referidas (como previsto no modelo comportamentalmente equivalente apresentado para as meta-redes).

Este procedimento deve ser repetido para todas as transições de entrada na superpágina.

Caso se utilize a característica de história, os procedimentos referidos no parágrafo anterior são insuficientes. Nesse caso torna-se necessário considerar individualmente os dois grupos de transições síncronas de entrada: as especificadas pelo projectista (num total de  $N$ ) e as geradas para suporte ao conceito de história (num total de  $M$ , igual ao número de lugares simples). Do primeiro grupo, analisado no parágrafo anterior, resulta o disparo de uma transição (por definição) em cada activação da sub-rede. No entanto, no segundo grupo, o número de disparos é dependente da sub-rede especificada, em particular do estado de marcação quando da última desactivação. Deste modo, para que a sua tradução possa ser realizada em termos de um conjunto de transições como realizado no parágrafo anterior, importa caracterizar todas as situações possíveis de disparo correspondentes a reactivações da sub-rede. É, no caso geral, tarefa que se afigura só possível pela análise prévia da sub-rede e de reduzido interesse prático, dado a explosão de combinações daí resultante transformar o modelo plano resultante de dimensão não compatível com as capacidades computacionais (normalmente) disponíveis. Para além disso, o modelo proposto é “demasiado de baixo-nível” para

acomodar convenientemente qualquer marcação colorida verificável na super-rede. Por esses motivos, o trabalho que permita identificar os procedimentos gerais para a inserção de super-redes com atributo de história foi deixado em aberto, para desenvolvimento em trabalhos futuros.

Contudo, para os casos em que o modelo da subpágina esteja monomarcado em todas as situações possíveis de desactivação é possível obter um modelo equivalente simples, através dos seguintes procedimentos sistemáticos. Como caso particular, refiram-se as máquinas de estados finitos, em que se satisfaz este condicionamento. Nestas situações, o número de conjuntos possíveis de disparo de transições síncronas de entrada quando de reactivações da sub-rede é, no máximo, igual ao número de lugares simples (considerando que alguma vez todos os lugares podem ser marcados). Nesta situação, as  $M$  transições do segundo grupo de transições (associadas à implementação da história) devem ser somadas as  $N$  transições especificadas pelo utilizador, no sentido de determinar o número de transições no modelo plano que modelam uma transição de entrada na superpágina. As regras apresentadas em parágrafo anterior para a fusão de transições, aplicam-se também neste caso, considerando que o lugar “super-rede nunca activada” é utilizado como lugar de entrada das  $N$  transições associadas às especificadas pelo projectista e o lugar “super-rede não activa” para as  $M$  transições de história.

Para os casos em que se pretenda modelar actividades solidárias, no sentido atrás referido de actividades com activação e desactivação conjunta e simultânea, importa analisar a obtenção do modelo comportamentalmente equivalente para um supernó a que esteja associado um conjunto de super-redes solidárias. Nesta situação, os procedimentos dos parágrafos anteriores serão alterados no sentido de permitir a marcação simultânea de todas as  $M$  super-redes solidárias. Considerando que a super-rede solidária  $SR_i$  dispõe de  $N_i$  transições síncronas de entrada especificadas, por cada transição de entrada no supernó serão geradas um número de transições igual ao produto  $N_1 * \dots * N_m$  (as considerações produzidas no parágrafo anterior com respeito ao suporte ao conceito de história são aplicáveis). Os arcos de entrada provenientes da superpágina são fornecidos a todas as transições produzidas. Cada uma das transições geradas representa uma possibilidade de marcação do grupo de super-redes solidárias. Desta forma, cada uma delas será fundida com  $M$  transições síncronas de entrada, uma de cada super-rede, de acordo com a combinação associada à transição. As guardas de cada uma dessas transições resultam da agregação das guardas da transição da superpágina e de cada uma das guardas especificadas nas transições síncronas. Os arcos de saída dessas transições são semelhantes aos arcos de saída das transições síncronas de entrada associadas. Eventos associados à transição da superpágina serão associados a todas as transições produzidas. Este procedimento deve ser repetido para todas as transições de entrada no supernó. De modo coerente com o apresentado nos parágrafos anteriores, cada uma das transições fundidas considerará como lugar de entrada um lugar

de controlo de cada super-rede solidária, concretamente o lugar modelando “super-rede não activa” ou “super-rede nunca activada”, conforme adequado.

#### 4.3.4.b.3 Modelando a desactivação das sub-redes

No que se refere aos arcos de saída do supernó, envolvendo as transições síncronas de saída, duas situações devem ser consideradas:

- a primeira diz respeito à modelação da interrupção (quer forçada quer permitida) de um processo por um outro hierarquicamente superior, representada por uma transição ligada a arco de saída do supernó; no caso da modelação permitida, a saída  $Exit[n]$ , emitida pela sub-rede, é utilizada como evento associado à transição;
- a segunda situação diz respeito à modelação da conclusão normal da actividade, representada na sub-rede através de uma transição síncrona de saída com o atributo “*Exit*” e na superpágina através da utilização do evento *exit*.

Na primeira situação (modelação da interrupção), o modelo plano comportamentalmente equivalente é obtido através da utilização de um arco de entrada na transição em causa que parta do lugar modelando “super-rede activa”. Os eventos de entrada associados são mantidos enquanto que é adicionado o evento “*Stop*” aos eventos de saída associados ao seu disparo, que irá provocar a desmarcação dos lugares simples. A alteração da marcação da sub-rede responsável pela modelação do estado de actividade da super-rede é garantida através de arco de saída da transição de saída para o lugar modelando “super-rede não activa”. Este procedimento deve ser repetido para todas as transições de saída do supernó.

Para o caso particular de subpáginas monomarcadas é possível traduzir essas dependências explicitamente em termos do modelo RdP-R obtido, evitando a utilização do disparo sincronizado de transições, garantido através da comunicação por eventos de saída e entrada das várias transições envolvidas. Os procedimentos a utilizar são duais aos utilizados para as transições de entrada e apresentam-se seguidamente.

Considere-se uma transição  $t1$  de saída do supernó ao nível da superpágina; considere-se um conjunto de  $N$  transições síncronas de saída ao nível da subpágina, interligadas aos lugares simples e dispondo do atributo “*Stop*”. O modelo resultante da inserção da subpágina substitui a transição de saída  $t1$  por um vector de transições  $t1[1..N]$ , em que cada uma das transições do vector será fundida com uma transição síncrona de saída, considerando-se para todas elas arcos de saída semelhantes aos arcos de saída de  $t1$ . Os arcos da transição síncrona de saída são ligados à transição fundida. Eventos

associados à transição  $t_1$  serão associados a todas as transições  $t_1[1..N]$ . As guardas de cada uma das transições do vector resultam da agregação das guardas da transição  $t_1$  e de cada uma das guardas especificadas nas transições síncronas. O lugar modelando “super-rede activa” deve ser considerado como lugar de entrada para todas as transições fundidas referidas, bem como o lugar modelando “super-rede não activa” como lugar de saída.

Este procedimento deve ser repetido para todas as transições de saída na superpágina.

Para os casos em que se pretenda integrar a modelação de actividades solidárias, representadas através de um conjunto de super-redes solidárias monomarcadas, os procedimentos dos parágrafos anteriores serão alterados no sentido de permitir a desmarcação simultânea de todas as  $M$  super-redes solidárias. Considerando que a super-rede solidária  $SR_i$  dispõe de  $N_i$  transições síncronas de saída associadas à desmarcação dos  $N_i$  lugares simples da super-rede, por cada transição de saída do supernó serão geradas um número de transições igual ao produto  $N_1 * \dots * N_m$ . Cada uma das transições geradas representa uma possibilidade de desmarcação do grupo de super-redes solidárias. Desta forma, cada uma delas representa a fusão da transição de saída com  $M$  transições síncronas de saída, uma de cada super-rede, de acordo com a combinação associada à transição. Os arcos de entrada para estas transições são provenientes da combinação associada de lugares simples das super-redes. Os arcos de saída dessas transições são semelhantes aos arcos de saída da transição de saída do supernó. Eventos associados à transição da superpágina serão associados a todas as transições produzidas. De modo coerente com o apresentado nos parágrafos anteriores, cada uma das transições fundidas considerará os lugares “super-rede activa” de cada super-rede solidária como lugares de entrada, bem como os lugares “super-rede não activa” como lugares de saída.

Este procedimento deve ser repetido para todas as transições de saída do supernó.

No caso particular associado à modelação de interrupção permitida, o modelo comportamentalmente equivalente pode ser simplificado. Nestes casos o evento associado à transição de saída tem uma dependência da saída  $Exit[n]$ . A simplificação corresponde a modelar explicitamente essa dependência através da estrutura do grafo, resultando na remoção de todas as transições que, dependentes de  $Exit[n]$ , não recebam arcos dos lugares que provocam a sua activação.

Na segunda situação caracterizada no início desta subsecção, em relação à modelação da conclusão normal da actividade, representada através de uma transição de saída com referência a *exit* no evento associado ao nível da superpágina, o modelo plano comportamentalmente equivalente é obtido através da utilização de um arco de entrada na transição em causa que parta do lugar modelando “super-rede activa”. Esta transição é considerada como tendo prioridade inferior às transições eventualmente existentes associadas à modelação de interrupções. Os eventos de entrada associados (para além de

*exit*) são mantidos, enquanto que é adicionado o evento “*Stop*” aos eventos de saída associados ao seu disparo (provocando a desmarcação dos lugares simples). A alteração da marcação da sub-rede responsável pela modelação do estado de actividade da super-rede é garantida através de arco de saída das transições de saída para o lugar modelando “super-rede não activa”. Este procedimento deve ser repetido para todas as transições de saída do supernó que envolvam uma referência a *exit* no evento associado.

#### 4.3.4.b.4 Métodos de obtenção do modelo plano

A construção do modelo plano necessita de ser realizada utilizando uma estratégia ascendente, isto é, de inserção dos submodelos a partir dos níveis mais profundos da hierarquia. Dado que os procedimentos apresentados apenas estabelecem a ligação entre uma subpágina e a superpágina associada, é necessário que as subpáginas sejam inseridas nas superpáginas intermédias, de modo iterativo, garantindo que as relações hierárquicas se reflectirão no nível de topo.

Importa referir que a utilização isolada desta estratégia teria como consequência a impossibilidade de utilização de referências recursivas de supernós (a exemplo do verificado com os macronós na decomposição horizontal).

O suporte à utilização de referências recursivas é conseguido através da análise prévia das características comportamentais das várias superpáginas, no sentido de determinar as condições específicas de evocação de cada uma das super-redes utilizadas, de modo semelhante ao realizado com as meta-redes na decomposição vertical. Desta forma, a inserção da super-rede na superpágina onde é referida só se verifica quando o supernó é efectivamente marcado. Ou seja, desde que se garanta que a recursividade não é realizada infinitamente (correspondendo a um sistema não realizável fisicamente), a utilização de referências recursivas é permitida.

Deste modo, os procedimentos de construção do modelo plano podem ser caracterizados em duas fases:

- a primeira utilizando uma estratégia descendente e tendo por objectivo caracterizar as marcações dos supernós e formas de activação das super-redes (quais as transições de entrada utilizadas); o método de análise a utilizar basear-se-á na construção do espaço de estados associado à superpágina;
- a segunda baseada numa estratégia ascendente, na informação obtida na passagem descendente e nos procedimentos de modelação da activação e desactivação das super-redes, permitirá obter o modelo plano comportamentalmente equivalente.

Um método alternativo ao agora apresentado, já referido em capítulo anterior, baseia-se na integração dos espaços de estados associados aos vários submodelos. Como principal vantagem, refira-se a sua aplicação universal e sistemática, independentemente das características dos submodelos utilizados; como principal desvantagem, realce-se a potencial explosão do número de estados do modelo global. Os procedimentos associados a este método são caracterizados nas duas fases referidas nos parágrafos anteriores, alterando a segunda fase referida no sentido de obter o espaço de estados global por fusão dos espaços de estados associados a cada um dos submodelos (de modo semelhante à verificada para a análise de uma especificação com statecharts). Deste modo, em lugar da obtenção de uma RdP plana, obtem-se uma máquina de estados comportamentalmente equivalente à RdP plana mas, obviamente, de dimensão, em termos do número de nós e arcos, potencialmente muito superior à RdP.

#### 4.3.4.c Retorno à modelação de statecharts através de RdP-R

No sentido de ilustrar a aplicação desta forma de decomposição, retoma-se o exemplo atrás apresentado de modelação de statecharts através de RdP-R (ver Figura 4-40). O conjunto de regras de tradução apresentado recorrendo à utilização de metalugares é válido, considerando que as referências a “metalugar” são substituídas por “superlugar”.

O statechart utilizado é, novamente, o apresentado na Figura 4-40(a), a que corresponde o conjunto de páginas RdP-R apresentado na Figura 4-40(b) (considerando agora que se utiliza um superlugar em vez de um metalugar). Os modelos comportamentalmente equivalentes correspondentes a cada uma das páginas, utilizáveis para a sua execução, apresentam-se na Figura 4-43. O modelo plano resultante da inserção dos modelos das super-redes B e C na página de topo, comportamentalmente equivalente ao statechart da Figura 4-40(a), apresenta-se na Figura 4-44. De notar que a única transição de entrada foi “desdobrada” em três transições (t1a , t1b e t1c), dado que em relação à super-rede B há a considerar um modo de activação e em relação à super-rede C há a considerar um modo de activação inicial, a que se adicionam duas possíveis reactivações através do mecanismo de história, sendo, assim, três o número de combinações possíveis. No que se refere à modelação da transição de saída, foi “desdobrada” em duas transições (t2a e t2b); das quatro transições potencialmente necessárias (2 transições síncronas de saída em cada uma das 2 super-redes), duas foram removidas dado não dependerem do lugar que activa a saída *Exit[1]*.

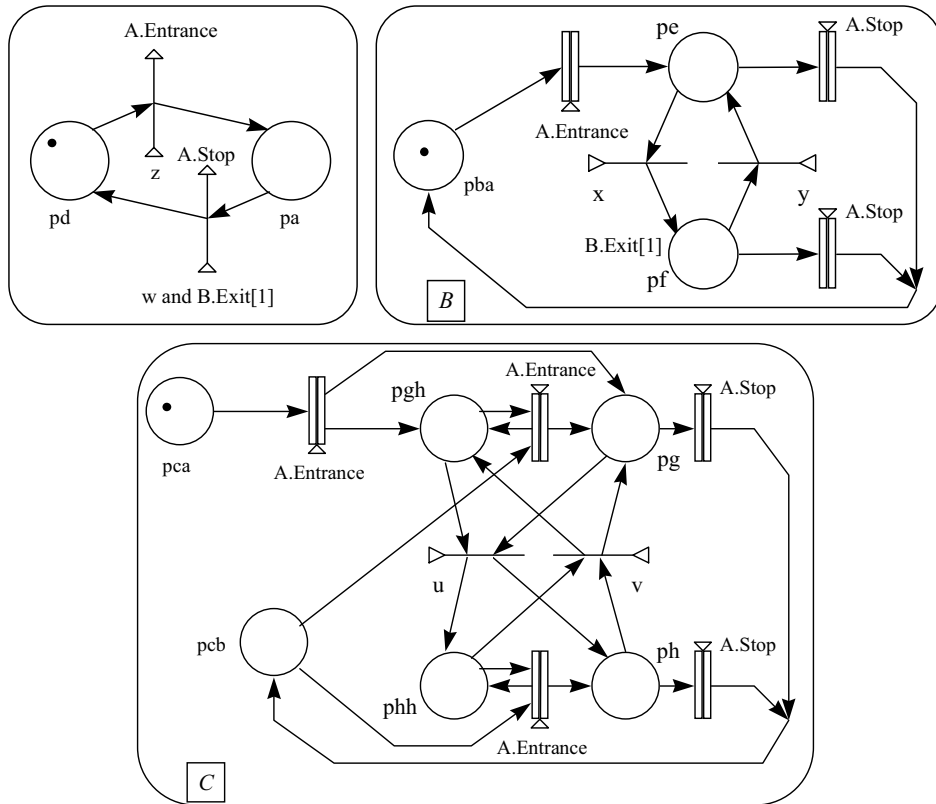


Figura 4-43 - Modelos de execução das páginas da Figura 4-40(b).

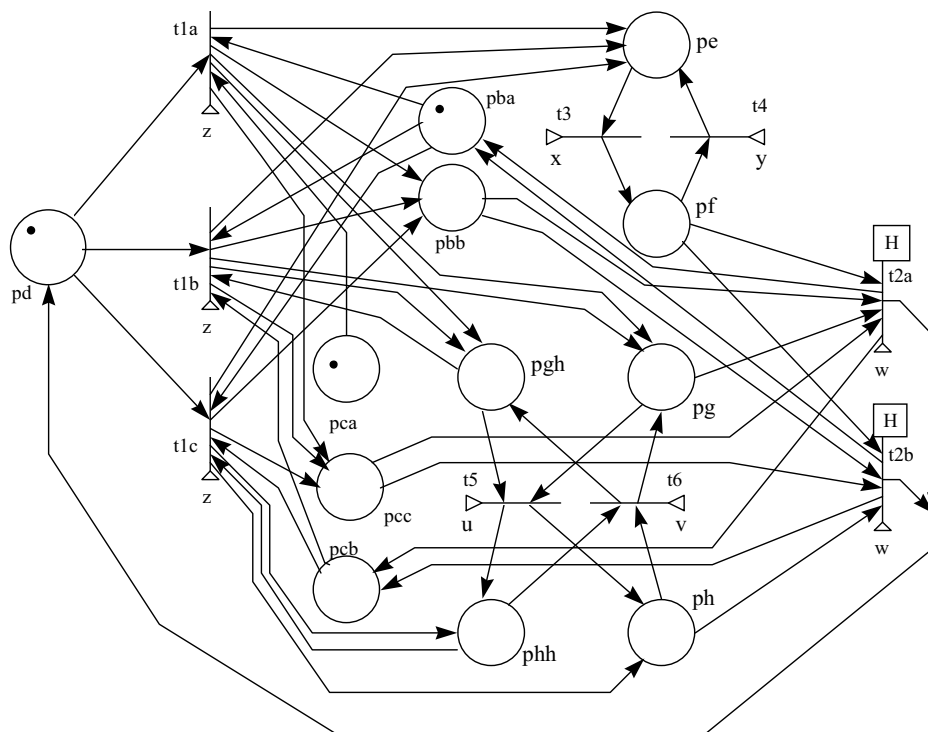


Figura 4-44 - Modelo plano comportamentalmente equivalente ao statechart da Figura 4-40(a).

Obviamente que o modelo plano apresentado não se destina a ser especificado (ou editado) directamente pelo projectista, sendo, de preferência, resultado de um processo automático de tradução

da especificação em statechart para RdP-R. Para melhorar a legibilidade da RdP-R, apresenta-se na Figura 4-45, uma sub-rede em que foram removidos os lugares das sub-redes de controlo das duas super-redes presentes, tornando fácil a identificação dos modelos associados às super-redes (uma dispondo do atributo de história), bem como das transições de entrada e saída.

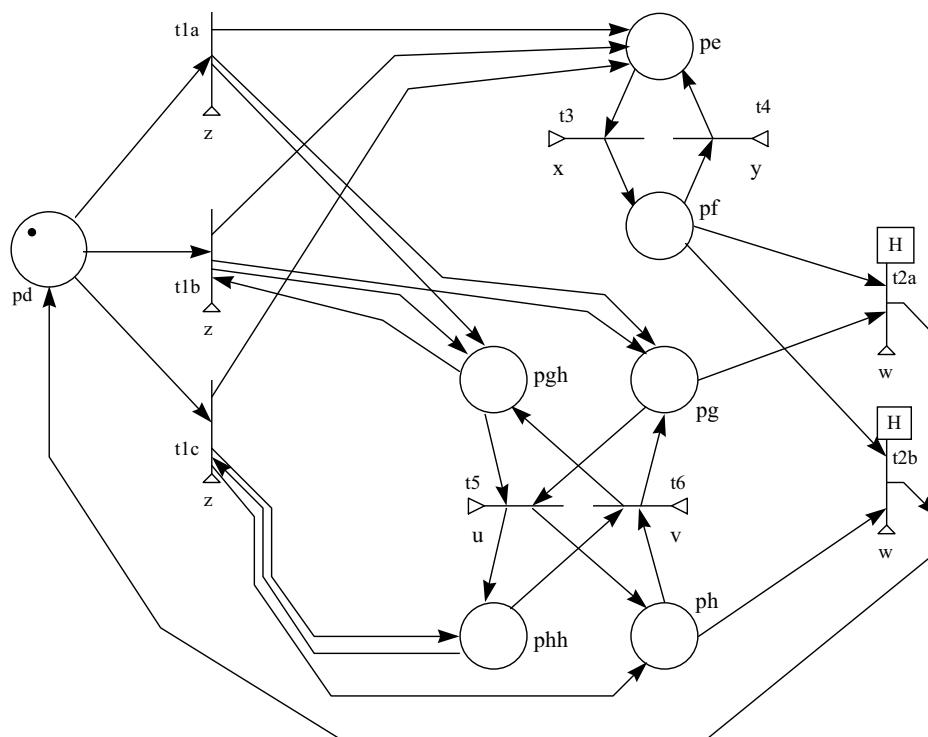


Figura 4-45 - Submodelo relativo à RdP da Figura 4-44.

Na Figura 4-46 apresenta-se a máquina de estados associada à RdP da Figura 4-44, obtida após tradução para C++ e compilação com a biblioteca desenvolvida em [Barros, 96].

De referir que, a máquina de estados apresentada na Figura 4-46, obtida a partir da RdP-R da Figura 4-44, equivalente ao statechart da Figura 4-40(a), contem sete estados, enquanto que o espaço de estados do statechart inicial apenas tem seis estados (como apresentado na Figura 4-41(b), obtida para o caso dos eventos de entrada serem mutuamente exclusivos, mas que se mantem em termos do número de estados para o caso geral de eventos não mutuamente exclusivos). Esta discrepância deve-se ao facto dos procedimentos propostos de tradução de statecharts em RdP-R não estarem otimizados para o caso de máquinas de estado, sendo gerado um estado inicial que não mais será visitado, mas que é equivalente a um outro estado do grafo, podendo com ele ser fundido. No caso do exemplo, os estados S1 e S6 da Figura 4-46 são equivalentes; a sua fusão permite obter a máquina de estados da Figura 4-47, semelhante à apresentada na Figura 4-41(b) (permitindo, no entanto, ocorrência simultânea de entradas).

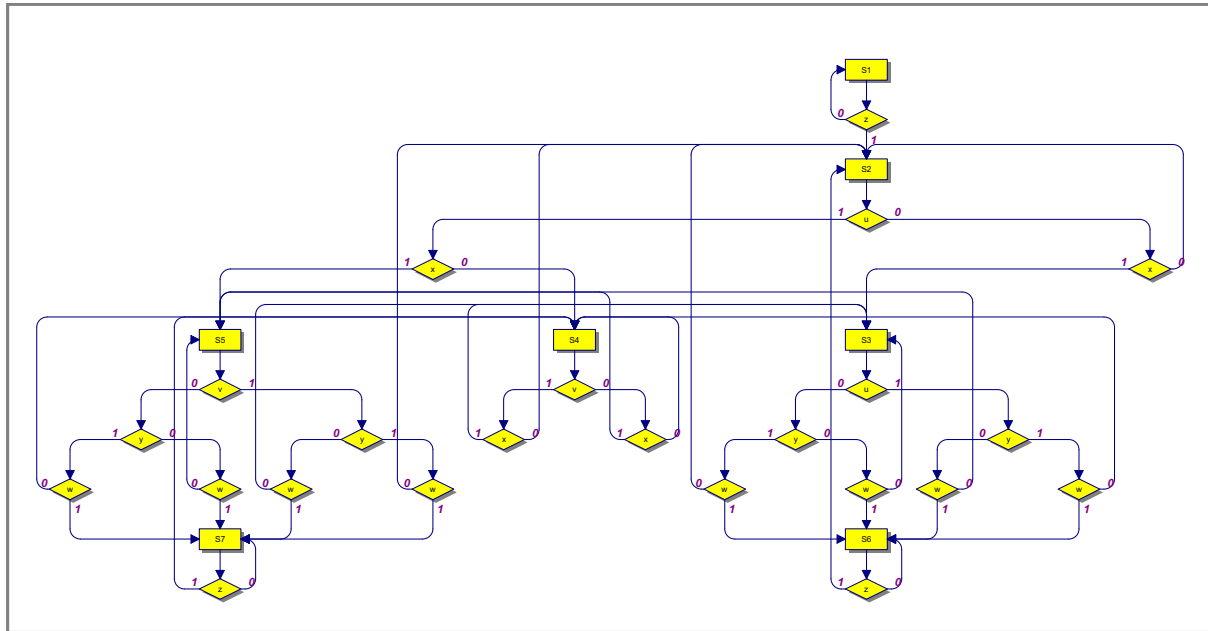


Figura 4-46 - Máquina de estados associada à RdP-R da Figura 4-44.

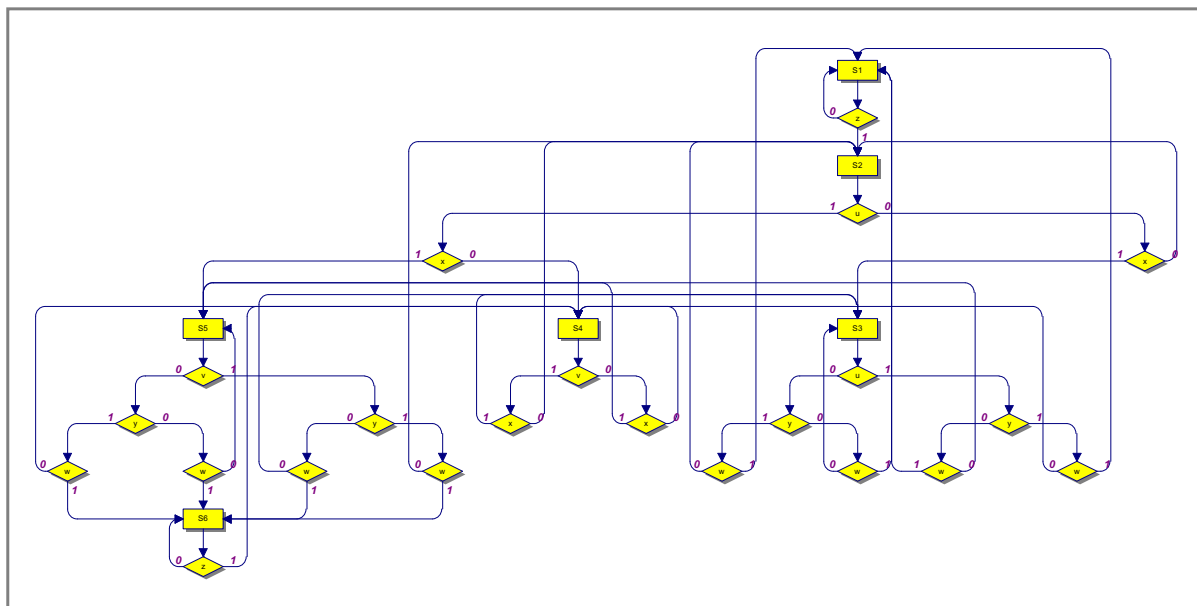


Figura 4-47 - Máquina de estados resultante da apresentada na Figura 4-46.

#### 4.4 Exemplo de aplicação

No sentido de ilustrar as técnicas propostas neste capítulo, recorre-se a um exemplo “tradicionalmente” utilizado na bibliografia da área, por exemplo em [Peterson, 77] [Reisig, 82], bem como em textos didáticos [Gomes, 96]. A Figura 4-48(a) apresenta uma RdP modelando, simplificada, um sistema produtor-consumidor.

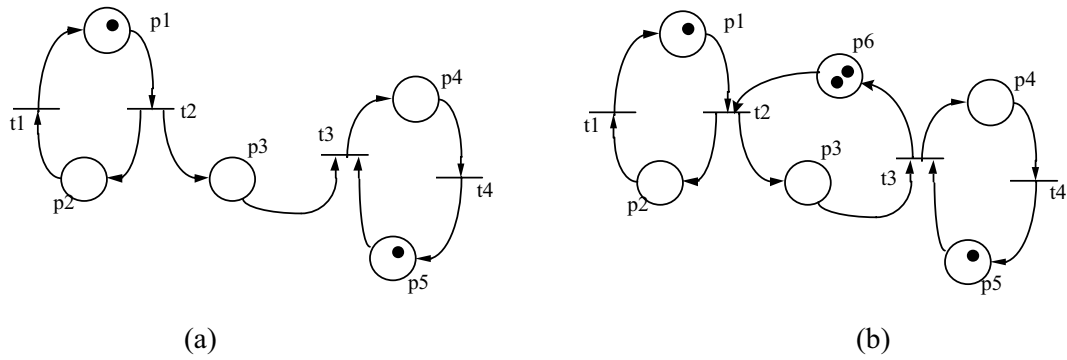


Figura 4-48 - Sistema produtor-consumidor:

a) armazém com capacidade infinita; b) armazém com capacidade igual a dois.

O ciclo de produção é modelado pelos lugares p1 e p2 e pelas transições t1 e t2, enquanto o ciclo de consumo agrupa os lugares p4 e p5 e as transições t3 e t4; o lugar p3 modela um armazém com uma capacidade de armazenamento infinito (isto é, é possível que o processo de produção ocorra infinitas vezes sem que o processo de consumo seja executado). A modelação de um armazém com capacidade finita (isto é, realizável), é possível através da utilização de um lugar complementar a p3.

A Figura 4-48(b) apresenta uma RdP satisfazendo à condição de armazém com capacidade igual a dois.

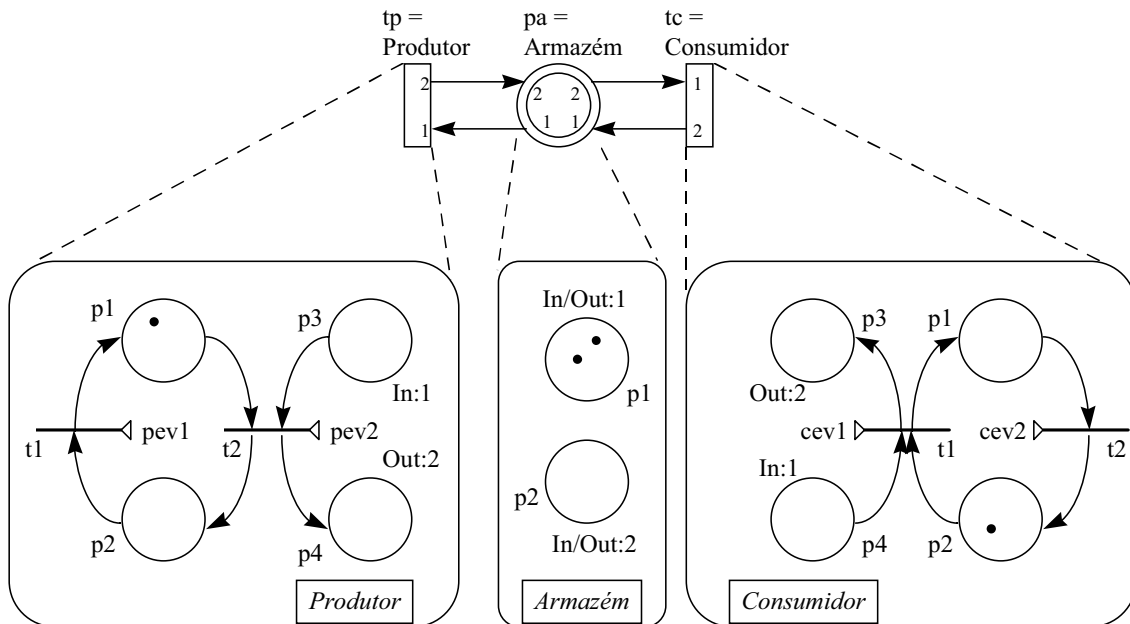


Figura 4-49 - Sistema produtor-consumidor:  
representação hierárquica com dois níveis.

### 4.4.1 Utilização de macronós

Uma estratégia óbvia de subdividir o modelo, baseia-se na divisão em produtor, consumidor e armazém. Dado que os dois primeiros submodelos representam acções e o terceiro representa um estado, uma representação compactada natural recorre à utilização de duas macrotransições e um macrolugar para modelar os submodelos referidos, como utilizado em [Barros, Gomes & Steiger-Garção, 97]. A Figura 4-49 apresenta a caracterização hierárquica correspondente. O modelo plano associado à Figura 4-49 foi apresentado na Figura 4-48(b). Embora a estrutura do consumidor e do produtor sejam semelhantes, conduzindo a macro-redes equivalentes, optou-se pela sua representação individualizada.

```
net ProdCons
macro place Armazem arm();
macro transition Produtor prod (place p3 = arm.p1, place p4 = arm.p2);
macro transition Consumidor cons (place p3 = arm.p1, place p4 = arm.p2);
```

```
net Armazem
place<Int> p1 "" { 2 * MS<Int>(Int(1)) } [];
place<Int> p2 "" {} [];
```

```
net Consumidor

code {
  static Expression Mark() {
    return new MS<Int>(Int(1));
  }
}

event cev1();
event cev2();

port place<Int> p3;
port place<Int> p4;

place<Int> p1 "" {} [];
place<Int> p2 "" {
  MultiSet<Int>(Int(1)) } [];

transition t1 "" {}
  in
    p2 { return Mark(); };
    p4 { return Mark(); };
```

```
  out
    p1 { return Mark(); };
    p3 { return Mark(); };
  [cev1] [] {};

transition t2 "" {}
  in
    p1 { return Mark(); };
  out
    p2 { return Mark(); };
  [cev2] [] {};

net Produtor

code {
  static Expression Mark() {
    return new MS<Int>(Int(1));
  }
}
```

```

event pev1();
event pev2();

port place<Int> p3;
port place<Int> p4;

place<Int> p1 "" {
    MultiSet<Int>(Int(1)) } [];
place<Int> p2 "" {} [];

transition t1 "" {}
in
    p2 { return Mark(); };
    
```

```

out
    p1 { return Mark(); };
    [pev1] [] {};

transition t2 "" {}
in
    p1 { return Mark(); };
    p3 { return Mark(); };
out
    p2 { return Mark(); };
    p4 { return Mark(); };
    [pev2] [] {};
    
```

Figura 4-50 - Descrições das páginas da RdP da Figura 4-49.

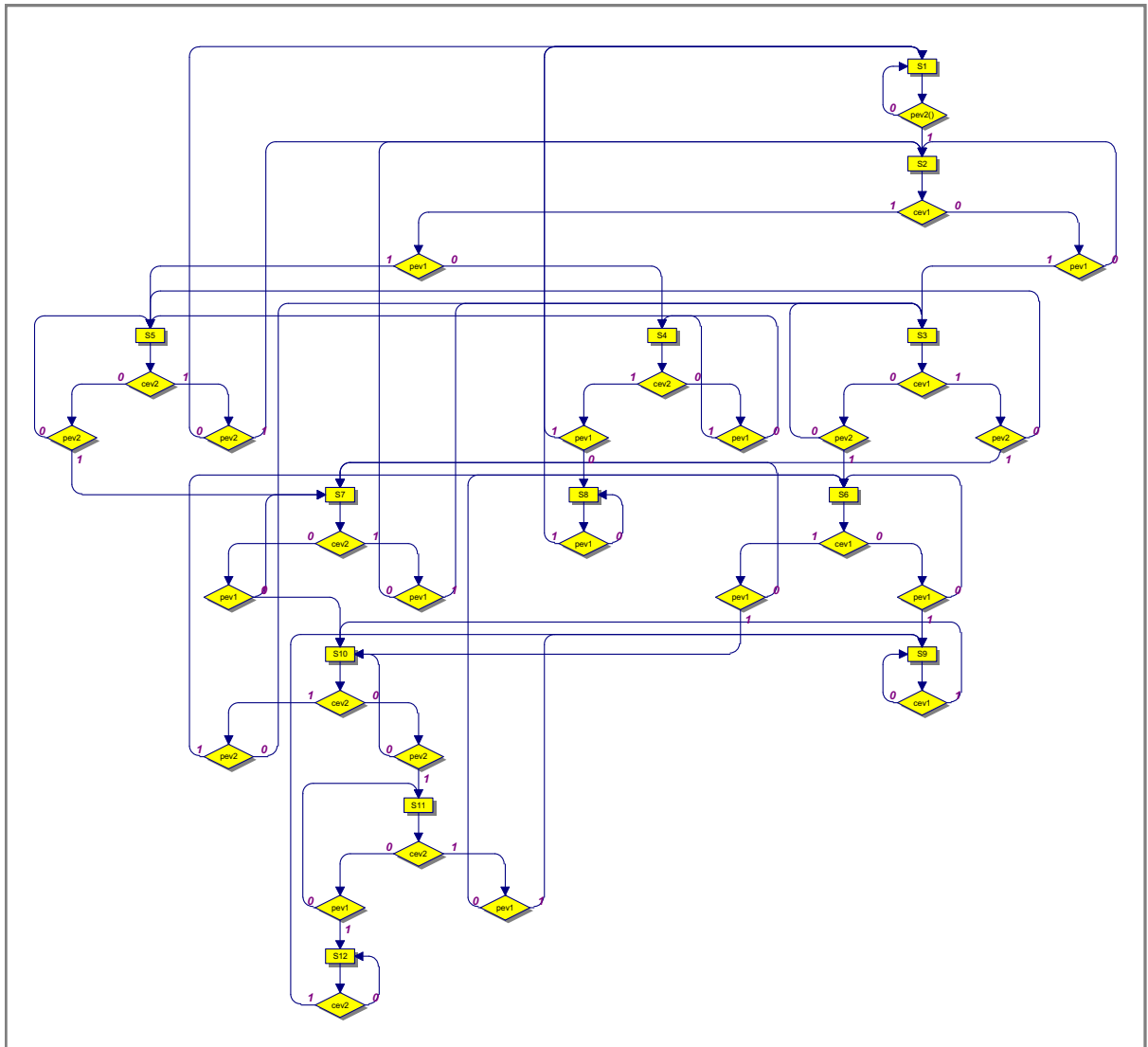


Figura 4-51 - Máquina de estados associada à RdP da Figura 4-48(b) e da Figura 4-49.

Na Figura 4-50 apresentam-se as várias páginas utilizadas para representar o modelo da Figura 4-49, enquanto que a Figura 4-51 apresenta a máquina de estados associada, contendo 12 estados.

Partindo deste modelo produtor-consumidor apresentado, serão analisadas várias situações que, embora se mantenham ao nível do exemplo didáctico, permitirão evidenciar algumas das potencialidades dos mecanismos de estruturação propostos.

#### 4.4.2 Utilização de representação estruturada baseada em vectores

Uma primeira situação a analisar, corresponde à expansão do sistema anterior a dois consumidores. Uma solução comum para a modelação do sistema descrito, recorre à utilização de atributos coloridos para o submodelo associado ao consumidor (o que impõe a sua alteração). A solução a analisar recorre à representação estruturada através de vectores, que tem como principal vantagem o facto de não necessitar de alterar os submodelos iniciais.

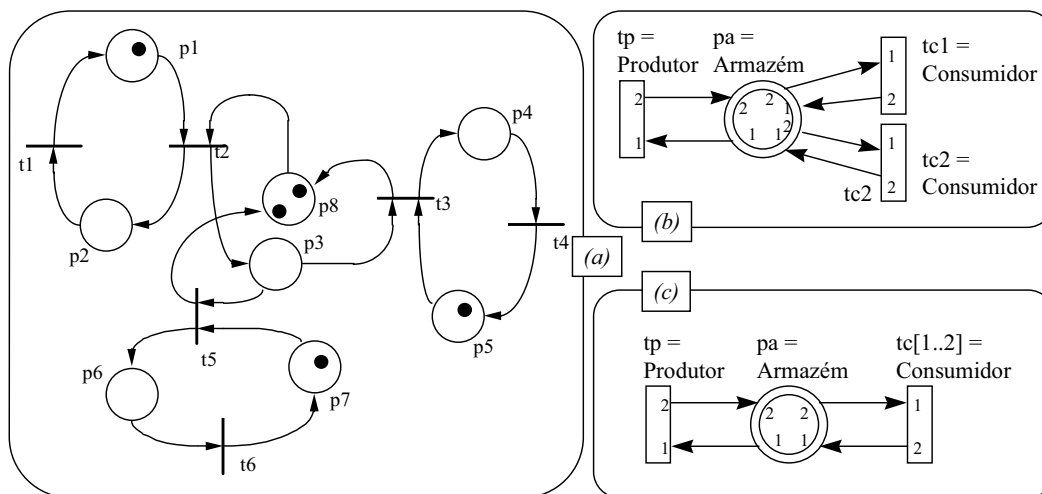


Figura 4-52 - Sistema com um produtor e dois consumidores com um armazém com capacidade igual a dois: a) modelo plano; b) recurso a macronós; c) recurso a macronós com estrutura.

O modelo plano correspondente ao sistema com um produtor e dois consumidores é apresentado na Figura 4-52(a); o ciclo contendo  $p1-p2$  modela o produtor, enquanto os ciclos contendo  $p4-p5$  e  $p6-p7$  modelam os dois consumidores; o lugar  $p3$  e o seu lugar complementar  $p8$  modelam o armazém com capacidade igual a dois (neste caso). A situação de conflito entre as transições  $t3$  e  $t5$  em relação à marcação de  $p3$  não será abordada neste caso (nem nos casos a analisar posteriormente); é suposto, para simplificar a exposição, que os eventos associados a essas transições possam resolver o conflito (sendo mutuamente exclusivos) ou que existam prioridades associadas às transições.

Uma primeira solução a analisar, baseada na atitude seguida no caso de um produtor - um consumidor, considera as mesmas macro-redes da Figura 4-49 evocadas a partir da nova superpágina,

como representado na Figura 4-52(b). As alterações a introduzir devidas ao novo consumidor limitam-se à adição de uma linha na página de topo, como apresentado na Figura 4-53. O espaço de estados do novo sistema comporta 24 estados, como espectável.

```
net ProdCons
macro place Armazem arm();
macro transition Produtor prod (place p3 = arm.p1, place p4 = arm.p2);
macro transition Consumidor cons1 (place p3 = arm.p1, place p4 = arm.p2);
macro transition Consumidor cons2 (place p3 = arm.p1, place p4 = arm.p2);
```

Figura 4-53 - Acomodar a introdução de um novo consumidor.

Considere-se, agora, o caso mais geral em que se podem considerar  $N$  subsistemas produtores e  $M$  subsistemas consumidores. Nesta situação, a utilização de  $N$  instâncias da macrotransição Produtor e de  $M$  instâncias da macrotransição Consumidor não corresponderá, quando  $N$  e  $M$  elevados, a uma solução clara (isto é, graficamente legível). Para esta situação, podem ser encaradas duas soluções:

- a primeira recorrendo à alteração das características do submodelo através da utilização de cores, com mais um atributo que possa representar a paginação dos vários modelos de produtores (ou consumidores) num único, topologicamente semelhante ao submodelo inicial;
- a segunda recorrendo à utilização de vectores de macronós, para o que não será necessário proceder a nenhuma alteração ao nível dos submodelos; a Figura 4-52(c) apresenta a solução aplicável para o modelo do sistema com um produtor e dois consumidores.

Fica, desta forma, evidenciada a potencial vantagem na utilização de vectores de nós para representar de modo compacto modelos gozando de características de repetição, suportando as atitudes de implementação incremental. Mais, comparando a solução apresentada com uma possível solução recorrendo à utilização de canais síncronos [Christensen & Hansen, 94], parece ser de concluir que o nível de abstracção conseguido é mais elevado (dado permitir modelos mais compactos).

#### 4.4.3 Utilização de metanós

Uma segunda situação a analisar, tomando como ponto de partida o sistema de um produtor e dois consumidores, endereça casos em que um ou mais nós da rede (isto é, das macro-redes) são substituídos por metanós, permitindo a modelação multinível de actividades hierarquicamente dependentes da actividade da superpágina.

Para comodidade de exposição, permitindo manter a complexidade do modelo gerível em termos da clareza que se pretende, considera-se que o lugar modelando a actividade de espera por novo produto a consumir (lugar p2 na macro-rede *Consumidor* da Figura 4-49) é substituído por um metalugar, tendo associado a meta-rede *Despacha*, como representado na Figura 4-54. A meta-rede *Despacha* modela uma sequência de acções responsáveis pelo “pós-processamento do produto consumido”, no caso ilustrado constituído por uma simples máquina de estados (permitindo a visualização dos modelos equivalentes).

Complementarmente, foi adicionada à transição t1 da macro-rede uma dependência no evento p2.exit, gerado pela meta-rede *Despacha* associada a p2. A adição desta dependência tem como consequência imediata a necessidade de atingir o estado de conclusão na meta-rede (marcação do lugar assinalado com Exit) para que a transição t1 da superpágina possa ocorrer. Como consequência particular, dado que a marcação inicial das meta-redes se realiza nos lugares p1, a possibilidade de um consumidor retirar marcas do armazém, fica condicionada pela existência de uma sequência de disparos na meta-rede que conduza à situação de conclusão; isto é, tudo se passa como se a meta-rede necessitasse de ser “inicializada”. Realça-se o facto desta característica (pouco “ortodoxa”) ser motivada exclusivamente pelo modo como os recursos de modelação foram utilizados para produzir o modelo.

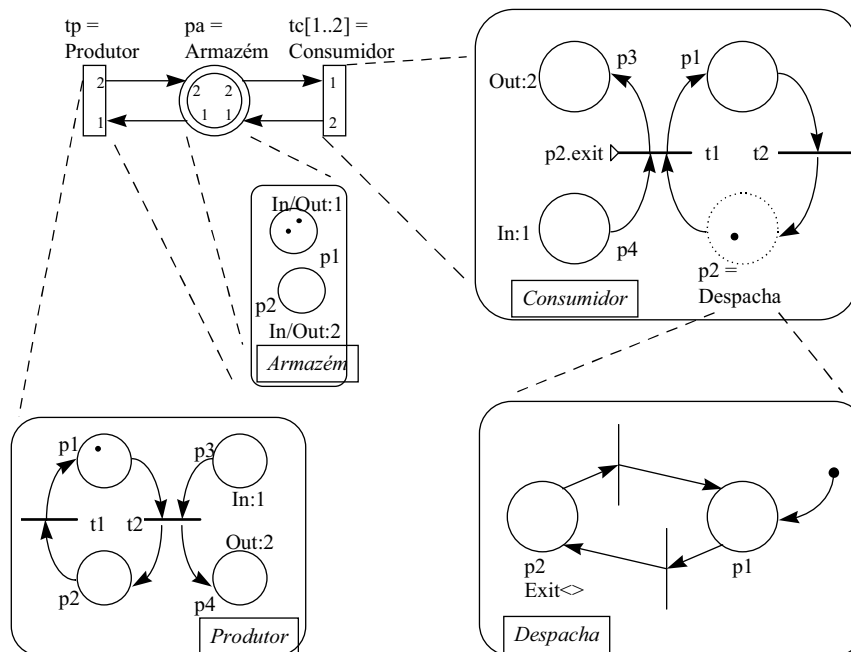


Figura 4-54 - Modelo hierárquico multinível de um sistema produtor - dois consumidores.

A execução do modelo apresentado na Figura 4-54 recorre à execução paralela de três submodelos; o primeiro resultado da inserção dos macronós na superpágina de topo e os outros dois responsáveis pela modelação das duas meta-redes *Despacha*. A aplicação dos procedimentos apresentados no corpo do presente capítulo permite obter os três submodelos referidos, que são apresentados na Figura 4-55.

De notar que, devido à marcação inicial do metalugar p2 na página *Consumidor*, a marcação inicial das meta-redes *Despacha[1]* e *Despacha[2]* é a resultante dos disparos das transições síncronas de entrada.

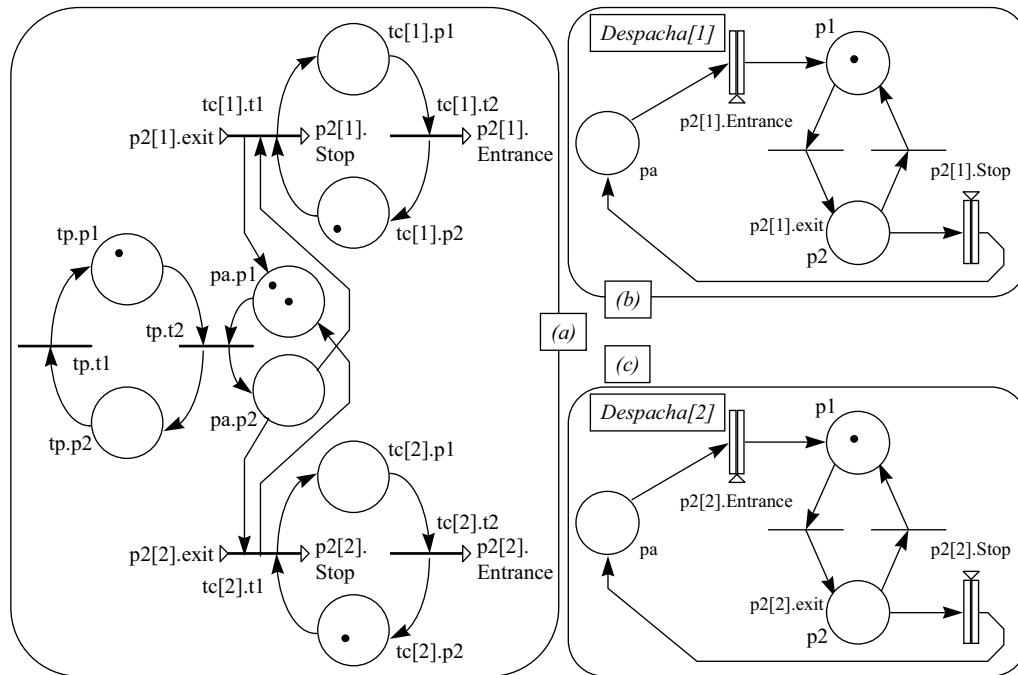


Figura 4-55 - Execução paralela do modelo hierárquico multinível da Figura 4-54.

A caracterização do espaço de estados associado à RdP da Figura 4-55(a) foi realizada na Figura 4-51; no que se refere às meta-redes *Despacha[1]* e *Despacha[2]*, o seu espaço de estado é coincidente com a RdP, dado a meta-rede representar uma máquina de estados monomarcada.

#### 4.4.4 Utilização de supernós

Uma terceira situação a analisar, utilizando o modelo hierárquico multinível de um sistema com um produtor e dois consumidores como ponto de partida, endereça casos em que se utilizem supernós. Para comodidade de exposição, considere-se que os metalugares utilizados na situação anterior são reinterpretados como superlugares.

O modelo da Figura 4-54 é válido nessa situação, considerando que o metalugar p2 da macro-rede *Consumidor* é substituído por um superlugar com idêntico submodelo associado. Estamos, desta forma, em presença de um caso claro de implementação de metanós com técnicas de macronó, como é característica dos supernós.

Como referido, de uma forma geral, podem ser utilizados dois métodos para obtenção de um modelo plano comportamentalmente equivalente:

- o primeiro método recorre à inserção das super-redes nas superpáginas onde são referidas, permitindo encontrar uma RdP-R comportamentalmente equivalente à RdP-RH inicial;
- o segundo método recorre à obtenção do espaço de estados global, por combinação dos espaços de estados parciais.

Recorrendo ao primeiro método referido para obtenção do modelo plano e com base nos procedimentos propostos, obtem-se a RdP-R plana da Figura 4-56. De modo a garantir a resolução automática dos conflitos presentes é necessário dispôr de três níveis de prioridade associados às transições envolvidas em conflitos (HH, H e L, respectivamente, em que HH representa o nível de prioridade mais elevado).

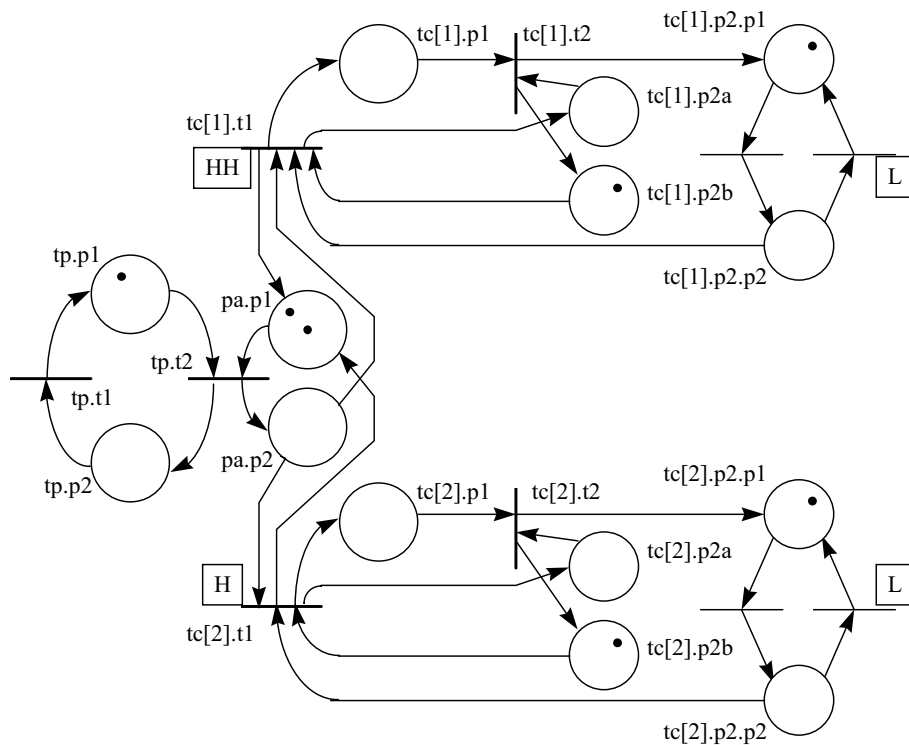


Figura 4-56 - RdP-R plana comportamentalmente equivalente à RdP-RH da Figura 4-54.

A sua representação textual é apresentada na Figura 4-57, tendo as referências aos eventos utilizados sido omitidas, por razões de legibilidade, nas Figura 4-56 e anteriores. A sua posterior tradução e compilação permite obter um espaço de estados com 54 estados, que não se apresenta, por razões que se prendem com a sua extensão (e interesse). Este valor é perfeitamente justificado pelo produto do número de estados de cada uma das sub-redes, em particular:

- produtor: 2 estados;
- armazem: 3 estados;

- consumidor: 3 estados (cada).

```

net Super

code {
  static Expression Mark() {
    return new MS<Int>(Int(1));
  }
}

event a();
event b();
event d();
event e();
event f();
event x();
event y();
event u();
event v();
event w();

place <Int> tpp1 "tpp1" {
  MS<Int>(Int(1)) } [];
place <Int> tpp2 "tpp2" {} [];
place <Int> pap1 "pap1" {
  2 * MS<Int>(Int(1)) } [];
place <Int> pap2 "pap2" {} [];
place <Int> tc1p1 "tc1p1" {} [];
place <Int> tc1p2b "tc1p2b" {
  MS<Int>(Int(1)) } [];
place <Int> tc1p2a "tc1p2a" {} [];
place <Int> tc1p2p1 "tc1p2p1" {
  MS<Int>(Int(1)) } [];
place <Int> tc1p2p2 "tc1p2p2" {} [];
place <Int> tc2p1 "tc2p1" {} [];
place <Int> tc2p2b "tc2p2b" {
  MS<Int>(Int(1)) } [];
place <Int> tc2p2a "tc2p2a" {} [];
place <Int> tc2p2p1 "tc2p2p1" {
  MS<Int>(Int(1)) } [];
place <Int> tc2p2p2 "tc2p2p2" {} [];

```

```

transition tpt1 "tpt1" {}
  in
    tpp2 { return Mark(); };
  out
    tpp1 { return Mark(); };
  [a] [] {} priority=3;

transition tpt2 "tpt2" {}
  in
    tpp1 { return Mark(); };
    pap1 { return Mark(); };
  out
    tpp2 { return Mark(); };
    pap2 { return Mark(); };
  [b] [] {} priority=3;

transition tc2t1 "tc2t1" {}
  in
    pap2 { return Mark(); };
    tc2p2p2 { return Mark(); };
    tc2p2b { return Mark(); };
  out
    pap1 { return Mark(); };
    tc2p1 { return Mark(); };
    tc2p2a { return Mark(); };
  [y] [] {} priority=2;

transition tc1t1 "tc1t1" {}
  in
    pap2 { return Mark(); };
    tc1p2p2 { return Mark(); };
    tc1p2b { return Mark(); };
  out
    pap1 { return Mark(); };
    tc1p1 { return Mark(); };
    tc1p2a { return Mark(); };
  [x] [] {} priority=1;

```

```

transition tc1t2 "tc1t2" {}
  in
    tc1p1 { return Mark(); };
    tc1p2a { return Mark(); };
  out
    tc1p2p1 { return Mark(); };
    tc1p2b { return Mark(); };
  [u] [] {} priority=3;

transition tc1p2t1 "tc1p2t1" {}
  in
    tc1p2p1 { return Mark(); };
  out
    tc1p2p2 { return Mark(); };
  [v] [] {} priority=3;

transition tc1p2t2 "tc1p2t2" {}
  in
    tc1p2p2 { return Mark(); };
  out
    tc1p2p1 { return Mark(); };
  [w] [] {} priority=3;

```

```

transition tc2t2 "tc2t2" {}
  in
    tc2p1 { return Mark(); };
    tc2p2a { return Mark(); };
  out
    tc2p2p1 { return Mark(); };
    tc2p2b { return Mark(); };
  [d] [] {} priority=3;

transition tc2p2t1 "tc2p2t1" {}
  in
    tc2p2p1 { return Mark(); };
  out
    tc2p2p2 { return Mark(); };
  [e] [] {} priority=3;

transition tc2p2t2 "tc2p2t2" {}
  in
    tc2p2p2 { return Mark(); };
  out
    tc2p2p1 { return Mark(); };
  [f] [] {} priority=3;

```

Figura 4-57 - Representação do modelo da Figura 4-56.

## 5. Modelação de técnicas de controlo inteligente

*Voz do Povo, voz de Deus*  
(*Provérbio popular, via erudita*)

*Voz do Povo, voz do Diabo*  
(*Provérbio popular*)

*A hora é incerta, mas a morte é certa*  
(*Provérbio popular*)

### ***Sumário do capítulo***

Pretende-se com o presente capítulo ilustrar a adequação das RdP-R para a modelação de formalismos normalmente referidos como de controlo inteligente, associados às áreas da Inteligência Artificial e da Inteligência Computacional, nomeadamente sistemas de regras de produção, regras com incerteza, regras difusas e de controlo difuso. O objectivo enunciado no capítulo inicial, que identificava a necessidade de dispôr de capacidade para integrar diferentes formalismos numa única especificação (em relação à qual fossem possíveis tarefas de verificação de propriedades e de execução) fica, deste modo, cumprido, no que concerne a este conjunto de formalismos.

Este capítulo é composto pelas seguintes secções:

*5.1 Introdução*

*5.2 Modelação de sistemas de regras*

*5.3 Modelação de controlo difuso*

*5.4 Aplicação a controladores programáveis*

*5.5 Retomando a resolução automática de conflitos*

A segunda secção inicia-se com um levantamento de alguns trabalhos vocacionados para a modelação de sistemas de regras de produção através de RdP de Alto-nível. A modelação de regras não-catóricas através de RdP-R é proposta (embora normalmente este tipo de regras não seja utilizado na modelação de sistemas e em controlo). A modelação de sistemas de regras com imprecisão e de regras de produção difusas abordar-se-á no restante desta secção.

As características específicas das RdP-R, permitindo integrar a modelação do fluxo de controlo com o fluxo de dados, têm benefícios específicos em termos das capacidades de modelação de regras difusas em que a agregação de informação pode ser realizada instantaneamente, suportando adequadamente a inferência difusa.

Na terceira secção abordam-se as questões específicas de modelação de controladores difusos através de RdP-R, sendo referidos os modelos de Mamdani-Assilian e de Takagi-Sugeno. Serão apresentadas as atitudes propostas de modelação de variáveis linguísticas difusas e das fases da inferência difusa e de codificação e decodificação difusas. Ilustrar-se-á a possibilidade de utilização de qualificadores associados às variáveis difusas. As questões da execução das especificações produzidas e da organização hierárquica de controladores serão abordadas no final da secção.

Os exemplos que se apresentarão ao longo deste capítulo serão extraídos da área dos edifícios inteligentes, tentando deste modo deixar claro o papel que essa área de aplicação desempenhou ao longo deste trabalho.

Na quarta secção, ilustra-se a execução de especificações RdP-R em plataformas de baixo custo, nomeadamente controladores programáveis de produção nacional, utilizando-se como exemplo um controlador simplificado de um sistema de ar condicionado.

Na última secção, retomar-se-á, brevemente, o tema da resolução automática de conflitos e as suas relações com a modelação de sistemas difusos.

## **5. Modelação de técnicas de controlo inteligente**

### **5.1 Introdução**

Como referido em capítulo anterior, a integração e execução concomitante de especificações de sub-modelos obtidos com recurso a formalismos distintos, é um dos objectivos dos trabalhos em que esta tese se insere.

As vantagens em permitir a especificação de sistemas através de uma formulação linguística são claras para os casos em que se torna inviável a obtenção de um modelo matemático rigoroso do comportamento do sistema. É nesse enquadramento que as técnicas de controlo inteligente devem ser consideradas, isto é, comportando-se de modo complementar em relação às técnicas clássicas de modelação de sistemas (e não competitivamente, como tão claramente é expresso pelo Prof. Lotfi Zadeh [Klir & Yuan, 96]).

A necessidade de gerir informação imprecisa ou incompleta, bem como a contradição, muitas vezes encontrada quando se parte de caracterizações linguísticas produzidas por diferentes especialistas (vejam-se, como exemplos, os provérbios populares que abrem este capítulo) reforça o interesse em considerar técnicas de raciocínio difuso, eventualmente estruturado de forma hierárquica. Esse será o rumo que se tentará seguir nas secções deste capítulo.

### **5.2 Modelação de sistemas de regras**

#### **5.2.1 Sistemas de regras de produção**

Encontram-se na bibliografia várias propostas para a implementação de máquinas de inferência para sistemas de regras baseadas em RdP de alto-nível [Murata & Zhang, 88] [Peterka & Murata, 89] [Murata, 89] [Li, 94] [Shimura, Lobo & Murata, 95]. Por outro lado, alguns trabalhos aplicam RdP e técnicas de controlo inteligente à programação de controladores programáveis [Atabakhche et al., 86]

[Martinez et al., 88] [Jafari & Boucher, 94] [Gomes & Steiger-Garção, 95a] [Gomes & Steiger-Garção, 95b] [Gomes & Steiger-Garção, 95c] [Gomes & Steiger-Garção, 95d].

A tradução de programas em lógica utilizando clausulas de Horn em RdP de alto-nível é uma tarefa sem grande complexidade aparente. Para uma regra do tipo

$$(\text{conclusão}) \leftarrow (\text{condição}_1), \dots, (\text{condição}_n) \quad n \geq 0$$

a seguinte representação alternativa poderá ser utilizada

$$\text{IF } (\text{condição}_1 \text{ AND } \dots \text{ AND } \text{condição}_n) \text{ THEN conclusão}^{31}.$$

Cada condição é modelada por um lugar; a presença (ausência) de marcação está associada à verificação (ou não) da condição. Desta forma, a estrutura de dados associada à marca guarda a informação linguística expressa pela condição ou conclusão. A regra propriamente dita é modelada através de uma transição com tantos lugares de entrada quantas as condições e com um lugar de saída, associado à conclusão.

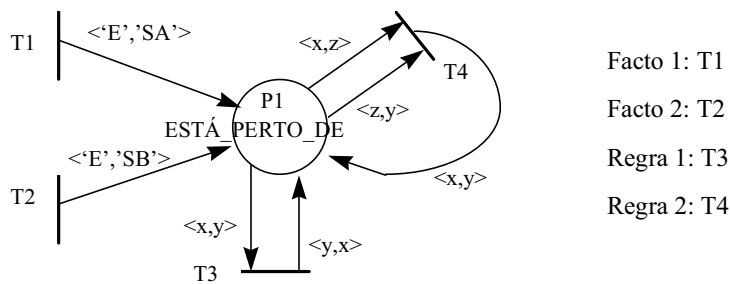
Quando  $n=0$ , a clausula modela um facto e é representado através de uma transição-fonte (ou transição-de-entrada). Uma pergunta à base de conhecimento pode ser modelada através de uma clausula sem conclusão, modelada através de uma transição-de-saída.

Exemplo 5.1: Considere-se o seguinte exemplo (num contexto de aplicação em actividades de gestão e monitoração em edifícios):

|          |   |
|----------|---|
| Facto 1: | Entrada está_perto_de SalaA   |
| Facto 2: | Entrada está_perto_de SalaB   |
| Regra 1: | IF Y está_perto_de X<br>THEN X está_perto_de Y                          |
| Regra 2: | IF X está_perto_de Z<br>AND Z está_perto_de Y<br>THEN X está_perto_de Y |

Os factos 1 e 2 modelam adjacências de espaços (Entrada, SalaA e SalaB). A regra 1 representa conhecimento comum, enquanto a regra 2 suporta o raciocínio permitindo obter novas conclusões.

De acordo com [Murata & Zhang, 88] [Murata, 89], a base de conhecimento apresentada pode ser representada pela RdP de alto-nível da Figura 5-1.



- Facto 1: T1
- Facto 2: T2
- Regra 1: T3
- Regra 2: T4

Figura 5-1 - A RdP associada ao Exemplo 5.1.

O processo de inferência pode ser representado pelo grafo de estados associado ao modelo RdP. A Figura 5-2 apresenta um grafo de estados parcial, associado com a sequência de disparo (T1, T2), (T1, T2, T3) e T4. O resultado do processo de inferência é a obtenção das conclusões “SalaA está\_perto\_de SalaB” e “SalaB está\_perto\_de SalaA”.

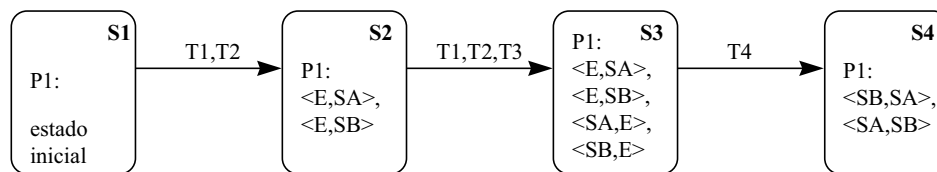


Figura 5-2 - Grafo de estados parcial associado ao modelo da Figura 5-1.

### 5.2.2 Regras não-categóricas

As regras até agora referidas são categóricas (como as que se utilizam comumente em sistemas baseados em regras). Embora não usual em ambientes de especificação de controlo de sistemas, é particularmente interessante considerar, para a modelação de sistemas complexos, regras não-categóricas, integrando a modelação explícita de excepções. Considere-se uma regra do tipo

IF antecedentes THEN conclusão  
UNLESS excepção

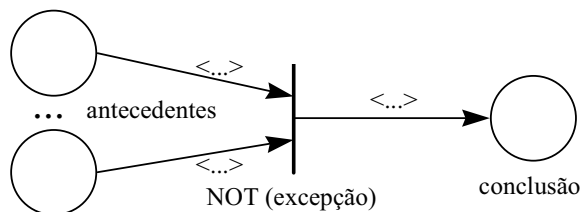


Figura 5-3 - Modelação de excepções em RdP-R.

<sup>31</sup> Optou-se pela representação das regras em inglês.

Considerando que a “exceção” pode ser avaliada logicamente, a sua modelação através de uma RdP de alto-nível, proposta em [Gomes & Steiger-Garção, 95e], baseia-se na sua integração na guarda associada à transição, como se apresenta na Figura 5-3.

Caso a exceção não possa ser avaliada como verdadeira ou falsa (seja uma proposição difusa, por exemplo), a sua modelação deverá ser realizada através de inclusão nos antecedentes da regra e os procedimentos a apresentar em secção posterior deverão ser utilizados.

Voltando ao Exemplo 5.1, no estado S3 da Figura 5-2 existem três vínculos possíveis:

- o vínculo de interesse (do ponto de vista da inferência) permite obter as conclusões “SalaA está\_perto\_de SalaB” e “SalaB está\_perto\_de SalaA”;
- os outros vínculos possíveis conduzem as conclusões triviais, nomeadamente “SalaA está\_perto\_de SalaA” e “SalaB está\_perto\_de SalaB”, e 2’ “Entrada está\_perto\_de Entrada”.

No sentido de evitar as conclusões triviais apresentadas, a regra 2 do Exemplo 5.1 deve ser alterada do seguinte modo, incluindo uma exceção:

```
Regra 2a:   IF X está_perto_de Z
            AND Z está_perto_de Y
            THEN X está_perto_de Y
            UNLESS X igual_a Y
```

A utilização de uma guarda “NOT(x==y)” associada à transição T4 da Figura 5-1 permite modelar esta nova regra, evitando o comportamento não desejado e, simultaneamente, eliminando os conflitos referidos.

### 5.2.3 Sistemas de regras de produção difusas

A necessidade de introduzir imprecisão na representação do conhecimento e de utilizar raciocínio aproximado determinou o aparecimento de novos paradigmas de representação de conhecimento, como os conjuntos difusos, e de inferência de informação [Zadeh, 96] [Yager, 84] [Klir & Yuan, 96].

Um extenso número de trabalhos utiliza modelos RdP para acomodar estes novos tipos de paradigmas, genericamente designados por sistemas de regras de produção difusas [Looney,88] [Chen, Ke & Chang, 90] [Gomes & Steiger-Garção, 92] [Cao & Sanderson, 93] [Chun & Bien, 93] [Bugarín & Barro, 94] [Pedrycz & Gomide, 94] [Yu, 95] [Gomes & Steiger-Garção, 95a] [Gomes & Steiger-

Garção, 95d] [Konar & Mandal, 96]. Um modelo RdP suportando o raciocínio em presença de inconsistência foi apresentado em [Murata, Subrahmanian & Wakayama, 91]

Em [Chen, Ke & Chang, 90] propõe-se a utilização de um factor de confiança associado a cada regra e a cada antecedente. Os antecedentes podem ser ligados através dos operadores AND e OR, enquanto se prevê a utilização de várias conclusões ligadas pelo operador AND. O cálculo dos factores de confiança associados às conclusões é realizado de acordo com a Tabela 5-1, onde  $y_k$  representa o factor de confiança associada a um antecedente ou conclusão e  $\mu_i$  representa o factor de confiança associado à regra.

| Tipo | Estrutura de regra   | Factor de confiança da conclusão             |
|------|--|--|
| 1    | CF= $\mu_i$<br>IF $d_{j1}$ AND ... AND $d_{jn}$ THEN $d_k$ | $y_k = \mu_i * \min (y_{j1}, \dots ,y_{jn})$ |
| 2    | CF= $\mu_i$ IF $d_j$<br>THEN $d_{k1}$ AND ... AND $d_{kn}$ | $y_{km} = \mu_i * y_j$ $m=1..n$              |
| 3    | CF= $\mu_i$<br>IF $d_{j1}$ OR ... OR $d_{jn}$ THEN $d_k$   | $y_k = \mu_i * \max (y_{j1}, \dots ,y_{jn})$ |

Tabela 5-1 - Computação dos factores de confiança das conclusões.

Em [Gomes & Steiger-Garção, 92] foi proposta a utilização de um novo atributo de cor para representar o factor de confiança. Assim, o factor de confiança é embebido na própria marca colorida presente no lugar representativo da relação linguística. A computação implícita ao cálculo dos vínculos associados a uma transição colorida permite proceder à determinação dos factores de confiança associados às marcas geradas pelo disparo da transição (ou seja, associados às conclusões da regra).

Em [Gomes & Steiger-Garção, 95a] e [Gomes & Steiger-Garção, 95d] foram propostas regras específicas de tradução para os diferentes tipos de regras apresentados, que se apresentam na Tabela 5-2. A regra propriamente dita é modelada por uma transição, com arcos de entrada ligados a lugares representativos dos antecedentes e arcos de saída ligados a lugares que representam as conclusões. Cada antecedente/consequente é modelado por uma marca colorida com dois atributos: o primeiro representando o valor linguístico e o segundo o factor de confiança associado. Os factos são representados por transições-fonte, gerando marcas pré-determinadas; alternativamente, podem ser utilizados lugares ligados a essas transições, cuja marcação inicial reflecte os factos conhecidos inicialmente.

| Tipo | Modelo associado | Tipo de regra e factor de confiança da conclusão   |
|------|------------------|--|
| 1    |                  | $CF = \mu_i$<br>IF $d_{j1}$ AND ... AND $d_{jn}$<br>THEN $d_k$<br>$y_k = \mu_i * \min (y_{j1}, \dots, y_{jn})$ |
| 2    |                  | $CF = \mu_i$<br>IF $d_j$<br>THEN $d_{k1}$ AND ... AND $d_{kn}$<br>$y_{km} = \mu_i * y_j \quad m=1..n$          |
| 3    |                  | $CF = \mu_i$<br>IF $d_{j1}$ OR ... OR $d_{jn}$<br>THEN $d_k$<br>$y_k = \mu_i * \max (y_{j1}, \dots, y_{jn})$   |

Tabela 5-2 - Tipos de regras, RdP-R associadas e factores de confiança de conclusão associados.

As estruturas das redes associadas aos tipos 1 e 3 de regras são idênticas (em [Chen, Ke & Chang, 90] utilizam-se estruturas distintas, embora seja necessário alterar as regras de disparo comuns das RdP, quando da determinação dos atributos das marcas geradas, permitindo a agregação dos factores de confiança associados aos vários antecedentes). Tenta-se, com isso, reflectir uma dependência semelhante em relação aos antecedentes. A distinção entre o tipo de operador utilizado para ligar os antecedentes reflecte-se unicamente na expressão que permite determinar o factor de confiança da conclusão.

Estes procedimentos de tradução propostos impõem a presença simultânea de marcas nos lugares de entrada da transição que modela a regra de produção difusa. Deste modo, a marcação inicial da rede deve garantir a existência de marcas cujos factores de confiança sejam inicializados a zero, caso a marca associada a determinado valor linguístico não esteja inicialmente disponível (por outras palavras, a marca existe sempre, sendo o factor de confiança utilizado para aferir do seu “grau de existência”).

Em [Gomes, 97] propõe-se um procedimento de tradução mais geral, com recurso às RdP-R. Considera-se, explicitamente, o caso de regras paralelas com conclusões semelhantes (formalmente idênticas à regra de tipo 3 apresentada). A fase de agregação das contribuições associadas aos diferentes antecedentes para o factor de confiança da conclusão é modelada explicitamente através de

uma transição síncrona. Essa transição síncrona é responsável pela agregação incremental de marcas associadas a um mesmo valor linguístico (ou relação linguística), permitindo a actualização incremental do factor de confiança associado a esse valor linguístico. A selecção do operador max para agregar a informação parcelar, dado permitir computação local, é coerente com o procedimento de tradução proposto.

| Tipo | Modelo associado | Tipo de regra e factor de confiança da conclusão   |
|------|------------------|--|
| 4    |                  | $CF = \mu_{i1} \text{ IF } d_{j1} \text{ THEN } d_k$<br>...<br>$CF = \mu_{in} \text{ IF } d_{jn} \text{ THEN } d_k$<br>$y_{mz} = \mu_{iz} * y_{jz} \quad z=1..n$<br>$y_k = \max (\mu_{i1} * y_{j1}, \dots, \mu_{in} * y_{jn})$ |
| 3bis |                  | $CF = \mu_i$<br>$\text{IF } d_{j1} \text{ OR } \dots \text{ OR } d_{jn}$<br>$\text{THEN } d_k$<br>$y_{mk} = \mu_i * y_{jk} \quad k=1..n$<br>$y_k = \max (y_{k1}, \dots, y_{k2})$   |
| 5    |                  | $y_k = \max (y_{k1}, \dots, y_{k2})$   |

Tabela 5-3 - Agregação de factores de confiança.

Na Tabela 5-3 apresentam-se as regras de tradução associadas, permitindo modelação alternativa à proposta na Tabela 5-2 no que se refere a regras de tipo 3 e integrando as propostas de [Chen, Ke & Chang, 90] e [Gomes & Steiger-Garção, 95a] [Gomes & Steiger-Garção, 95d], garantido-se a satisfação das regras de disparo da classe de RdP-R.

A linha 5 da Tabela 5-3 apresenta o procedimento sistemático a aplicar em todos os lugares que possam receber múltiplas marcas associadas a valores ou relações linguísticas semelhantes, permitindo a sua agregação numa única marca.

Desta forma, a implementação de uma máquina de inferência baseada em RdP-R baseia-se no fluxo contínuo de marcas, isto é, na actualização do estado do modelo ao longo dos vários passos de análise.

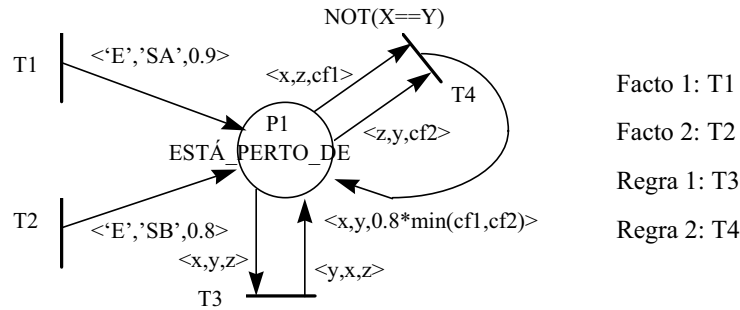


Figura 5-4 - A RdP-R associada ao Exemplo 5.2.

Exemplo 5.2: Considere-se a seguinte base de conhecimento, obtida com base no Exemplo 5.1, após a alteração da sub-secção anterior e introdução de factores de confiança:

Facto 1: CF=0,9 Entrada está\_perto\_de SalaA  
 Facto 2: CF=0,8 Entrada está\_perto\_de SalaB  
 Regra 1: CF=1 IF Y está\_perto\_de X  
 THEN X está\_perto\_de Y  
 Regra 2: CF=0,8 IF X está\_perto\_de Z  
 AND Z está\_perto\_de Y  
 THEN X está\_perto\_de Y  
 UNLESS X igual\_a Y

O modelo RdP-R da Figura 5-4 é obtido aplicando os procedimentos de tradução apresentados.

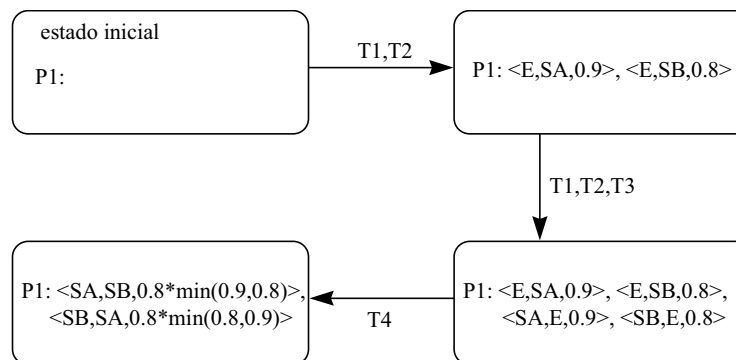


Figura 5-5 - Grafo de estados parcial associado à Figura 5-4.

De modo semelhante ao realizado para o Exemplo 5.1, o processo de inferência pode ser representado pelo grafo de estados associado ao modelo RdP-R. A Figura 5-5 apresenta um grafo de estados parcial, associado com a sequência de disparo (T1, T2), (T1, T2, T3) e T4, sendo o resultado do processo de inferência a obtenção das conclusões “CF=0,64, SalaA está\_perto\_de SalaB” e “CF=0,64, SalaB está\_perto\_de SalaA”.

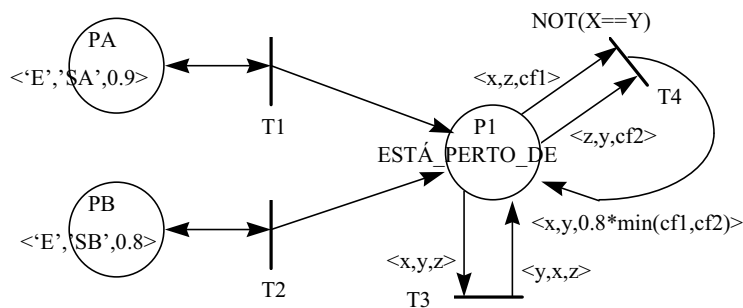


Figura 5-6 - Modelo alternativo equivalente para o Exemplo 5.2.

Uma visão alternativa do processo de inferência utiliza um modelo descompactado em que se representam explicitamente os lugares marcados em determinado passo de análise, permitindo verificar a evolução do modelo. Para encontrar esse modelo alternativo, considere-se o modelo da Figura 5-6, em que os factos são modelados através de lugares com uma marcação persistente interligados às transições-fonte atrás utilizadas. A Figura 5-7 apresenta a evolução do estado do modelo RdP-R da Figura 5-6.

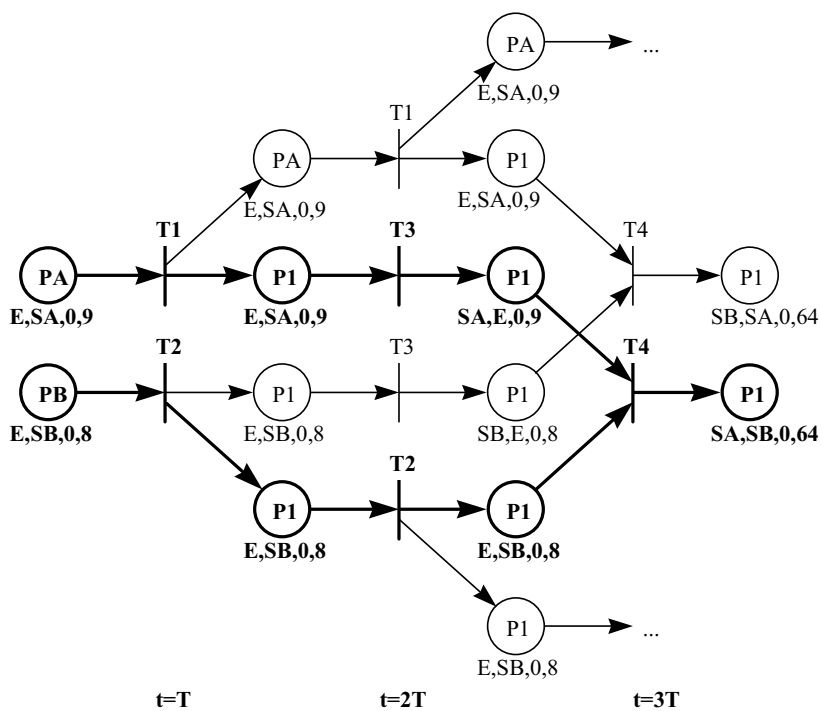


Figura 5-7 - Execução do modelo RdP-R descompactado.

Com base no modelo plano apresentado na Figura 5-7, é possível identificar a sub-rede responsável pela inferência em relação a objectivos específicos. Como exemplo, na Figura 5-7 intensifica-se a sub-rede responsável pela avaliação do factor de confiança associado ao objectivo “SalaA está\_perto\_de SalaB”.

De um modo geral, a sub-rede assim encontrada pode ser embebida no modelo que será utilizado em tempo de execução pelo controlador do sistema especificado, permitindo, deste modo (com as tarefas de compilação prévia já realizada), garantir tempos de resposta para os processos de inferência necessários às actividades de controlo do sistema.

Exemplo 5.3: Como exemplo adicional, considere-se a modelação de actividades de monitoração autónoma em edifícios desenvolvidas por um veículo autónomo (aqui denominado por Domot1). É suposto que algumas áreas do edifício dispõem de sistemas de identificação, onde o Domot1 se poderá auto-identificar. Como resultado, o sistema de identificação disponibiliza informação do tipo

Domot1 está\_em lugar CF= $\mu$

em que *lugar* poderá assumir os valores 'Entrada' e 'SalaA' e  $\mu$  representa o factor de confiança, dependente do tempo decorrido após a última identificação.

Considere-se a seguinte base de conhecimento que será utilizada concomitantemente com a do Exemplo 5.2:

Facto 3: CF=0,9 Domot1 está\_em Entrada  
 Facto 4: CF=0,6 Domot1 está\_em SalaA  
 Regra 3: CF=1 IF X está\_em Z  
 AND Z está\_perto\_de Y  
 THEN X está\_em Y

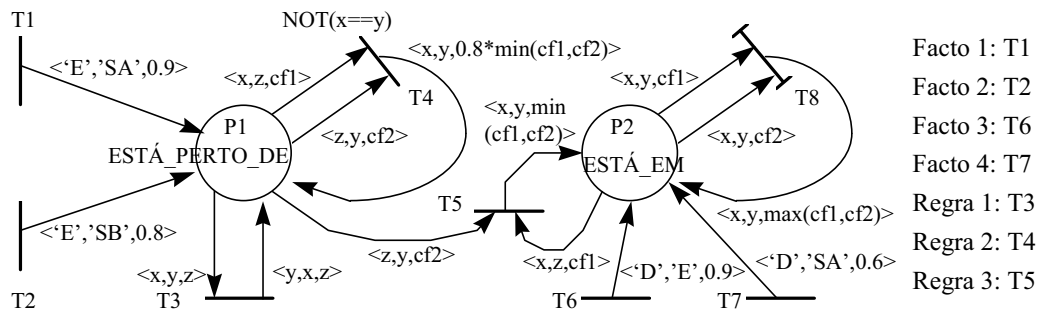


Figura 5-8 - A RdP-R associada ao Exemplo 5.3.

Os factos 3 e 4 são produzidos pelo sistema de identificação instalado no edifício. A RdP-R associada apresenta-se na Figura 5-8. A transição T8 é responsável pela agregação de marcas com informação linguística semelhante, permitindo a actualização do factor de confiança associado.

## 5.3 Modelação de controlo difuso

### 5.3.1 Modelando variáveis linguísticas difusas

A representação de valores linguísticas difusos pode ser encarada como uma generalização da representação de valores linguísticos aos quais se associam factores de confiança. Em [Gomes & Steiger-Garção, 95a] é utilizada uma estrutura de dados associada à marca (isto é, uma cor) perfeitamente similar à que foi utilizada na secção precedente, substituindo a representação do factor de confiança pelo valor de pertença.

Desta forma, a marca representativa de um determinado valor linguístico possui dois atributos:

- o primeiro é responsável por armazenar um identificador do valor linguístico, agregando o nome da variável e o conjunto difuso específico;
- o segundo é responsável por armazenar uma informação numérica, representando o nível de pertença actual.

### 5.3.2 Estrutura de um controlador difuso

Conceptualmente, a estrutura de um controlador difuso “clássico” é muito simples e constituída por vários blocos [Pedrycz, 93] [Kóczy, 95]:

- um bloco de entrada responsável pela aquisição de informação associada aos sinais de entrada de controlo; esta fase é normalmente referida na literatura em Inglês por “*fuzzification*” e não possui uma tradução pacífica para Português; alguns autores referem-se-lhe como “fuzzificação” ou “conversão numérico-linguística”; no presente texto, prefere-se a expressão “codificação difusa<sup>32</sup>”;
- um bloco responsável pela inferência e pelo raciocínio difuso; pode ser considerado como dividido em duas sub-fases:
  - fase de propagação das condições difusas, isto é, da inferência propriamente dita;
  - fase da agregação de informação, permitindo a combinação de conclusões semelhantes obtidas por caminhos de inferência diversos;

---

<sup>32</sup> Codificação difusa, tradução da expressão “fuzzy encoder”.

- um bloco responsável pela actuação, isto é associado aos sinais de saída; esta fase é normalmente referida na literatura em Inglês por “*defuzzification*”; em Português, alguns autores referem-se-lhe como “desfuzzificação” ou “conversão linguístico-numérica”; no presente texto, prefere-se a expressão “descodificação difusa<sup>33</sup>”.

Nas sub-secções seguintes cada um dos blocos referidos será analisado separadamente.

### 5.3.3 Codificação difusa

O bloco de codificação difusa é composto por uma transição-de-entrada para cada variável de entrada a considerar. Cada transição possui tantos arcos de saída quanto os valores linguísticos associados com a caracterização difusa da variável de entrada associada.

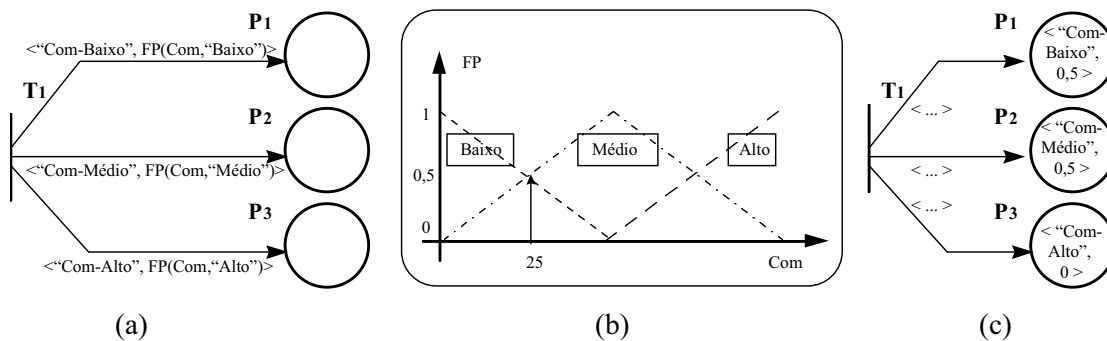


Figura 5-9 - Modelação da fase de codificação difusa: a) modelo RdP-R; b) conjuntos difusos associados à variável de entrada  $Com$ ; c) marcação gerada na situação de  $Com=25$ .

Cada marca gerada pelo disparo da transição-de-entrada possui dois atributos; o primeiro representando o valor linguístico propriamente dito e o segundo o valor de pertença actual. A Figura 5-9 apresenta um exemplo, ilustrando-se o vínculo obtido aquando do disparo da transição numa situação particular do sinal de entrada.

Como referido em capítulo anterior, desde que a cardinalidade do conjunto de vínculos possíveis para a transição de entrada seja finito, é possível encontrar o modelo de baixo nível comportamentalmente equivalente ao modelo apresentado. No caso presente, esta restrição traduz-se:

- num número finito de conjuntos difusos associados a cada variável de entrada; em situações de modelação típicas, o número de valores linguísticos é inferior ou igual a sete;

<sup>33</sup> Descodificação difusa, tradução da expressão “fuzzy decoder”.

- a resolução associada à codificação dos valores da função de pertinência necessita de ser finita; em situações de modelação típicas, utilizam-se 8 bits na codificação, sendo, no entanto, admissível um número inferior de níveis quantificáveis em muitas situações.

Deste modo, desde que estas condições sejam satisfeitas, é possível proceder à verificação formal da especificação produzida, recorrendo a métodos exaustivos, como a geração do espaço de estados.

### 5.3.4 Inferência difusa

O bloco de inferência é responsável pela determinação (ou actualização) de conclusões, com base no conjunto de regras especificado.

Tomando como referência o controlador proposto por Mamdani [Mamdani, 74] [Mamdani & Assilian, 75], a representação linguística de uma regra difusa é apresentada na Figura 5-10. A especificação do controlador difuso é constituída por um conjunto de regras difusas, como a apresentada na Figura 5-10, executadas concorrentemente.

```

IF    (    variável_linguística1    IS    valor_linguístico1j
      AND
      ...
      AND variável_linguístican    IS    valor_linguísticonk)
THEN variável_linguísticao    IS    valor_linguísticool
    
```

Figura 5-10 - Representação linguística de regra difusa.

Representação alternativa para o conjunto de regras, muito mais compacta e intuitiva quando o número de variáveis linguísticas de entrada é de dois, utiliza uma tabela em que as linhas e colunas representam os valores linguísticos possíveis para as variáveis de entrada e o conteúdo da tabela propriamente dito indica o valor linguístico da variável de saída. Na Figura 5-11 ilustra-se um caso em que a variável de entrada 1 (2) dispõe de n (m, respectivamente) valores possíveis. Esta representação tabular torna mais clara o grau de cobertura da especificação produzida (dado que podem existir combinações dos valores das variáveis de entrada para as quais não se indiquem conclusões).

|                                       |                                       |                                 |                                       |
|---------------------------------------|---------------------------------------|---------------------------------|---------------------------------------|
|                                       | <b>valor_linguístico<sub>11</sub></b> | ...                             | <b>valor_linguístico<sub>1n</sub></b> |
| <b>valor_linguístico<sub>21</sub></b> | ...                                   | ...                             | ...                                   |
| ...                                   | ...                                   | valor_linguístico <sub>ok</sub> | ...                                   |

|                                       |     |     |     |
|---------------------------------------|-----|-----|-----|
| <b>valor_linguístico<sub>2m</sub></b> | ... | ... | ... |
|---------------------------------------|-----|-----|-----|

Figura 5-11 - Representação tabular de um controlador difuso.

A avaliação de cada uma das condições presentes na regra é realizada na fase de codificação difusa. A modelação e execução de cada uma das regras a utilizar é realizada de forma semelhante à apresentada na secção anterior, sendo o cálculo dos valores de pertença associados aos valores linguísticos da conclusão realizado de forma semelhante à apresentada para o cálculo dos coeficientes de confiança, isto é, utilizando o operador *min* para determinar o valor de pertença da conclusão de uma regra, e o operador *max* para agregar valores de pertença associados a um mesmo conjunto difuso, obtidos através de regras avaliadas paralelamente.

Exemplo 5.4: Como exemplo, a ser retomado posteriormente, considere-se o seguinte sub-conjunto de regras associadas a um controlador difuso:

IF Temperatura IS Zero AND Comando IS Alto THEN Ventoínha IS Média

IF Temperatura IS Negativa AND Comando IS Alto THEN Ventoínha IS Alta

em que:

- *Temperatura* e *Comando* representam variáveis linguísticas de entrada,
- *Ventoínha* representa uma variável linguística de saída,
- *Alto*, *Média* e *Alta* representam valores possíveis para as referidas variáveis.

O modelo RdP-R associado apresenta-se na Figura 5-12. Os nomes das variáveis são referidos de forma abreviada.

O conjunto de lugares à esquerda (P1 a P3) armazena as marcas associadas ao processo de codificação difusa das entradas, enquanto que a segunda coluna (P4 e P5) recebe as marcas resultantes da inferência.

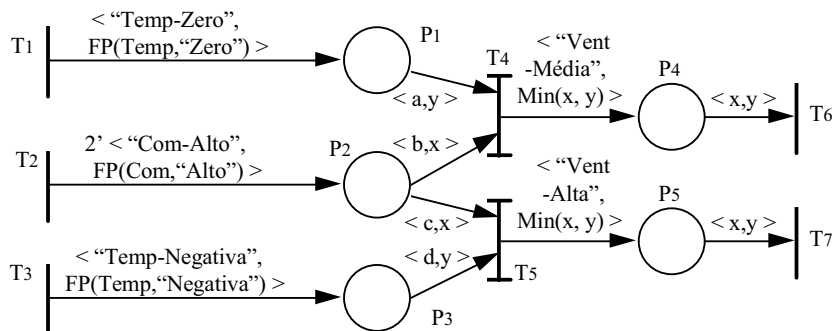


Figura 5-12 - Modelo RdP-R associado ao Exemplo 5.4.

Caso várias marcas com o mesmo valor linguístico fossem produzidos como resultado deste processo de inferência (situação que não se verifica no exemplo apresentado), tornava-se necessário agregar os seus valores de pertinência através de uma transição síncrona, a exemplo do realizado na secção anterior (ver Tabela 5-3 - linha 5).

De notar, ainda em relação ao exemplo apresentado, que, quando num conjunto de regras, se necessitar de avaliar em mais do que uma regra o valor de pertinência associado a um determinado valor linguístico, torna-se necessário que a transição-de-entrada responsável pela codificação difusa gere o número de marcas (semelhantes entre si) necessárias para que não se gerem situações de conflito, isto é, que sejam geradas tantas marcas quantas as regras que necessitam de avaliar esse valor linguístico. No exemplo apresentado, essa situação é verificada em relação ao antecedente “Comando IS Alto”, com a geração de duas marcas iguais.

Este procedimento de replicação de marcas com o objectivo de evitar conflitos ao longo do processo de inferência é válido (e necessário) também nas situações de modelação descritas na secção anterior.

No exemplo apresentado, a tradução de cada uma das regras foi realizada utilizando transições síncronas. No entanto, a utilização de transições assíncronas também é válida, estando a escolha associada ao modo como se pretenda evoluir no processo de inferência. Em sub-secção posterior retomar-se-á o assunto.

### 5.3.5 Utilização de qualificadores

Um expediente largamente utilizado nas tarefas de modelação de sistemas recorre à utilização de qualificadores associados aos vários valores linguísticos definidos [Zimmermann, 85]. Um caso típico é o de utilização de “intensificadores”, de que “muito” é, provavelmente, o exemplo mais comum.

Baseiam-se na transformação da função de pertinência definida para o conjunto difuso A através de uma função do tipo:

$$m_p(A) = \sum \mu_A^p(x)/x$$

em que:  $m$  é o qualificador;

$\mu$  é o valor de pertença inicial;

$p$  é um parâmetro especificado para cada tipo de qualificador.

Como valores típicos para  $p$ , refiram-se  $p=2$  obtendo-se o qualificador *muito* ( $\mu^2$ ) e  $p=1/2$  obtendo-se o qualificador *mais-ou-menos* ( $\mu^{1/2}$ ).

Como exemplo da sua utilização, considere-se a seguinte regra:

IF Temperatura IS Muito Negativa AND Comando IS Alto THEN Ventoinha IS Alta

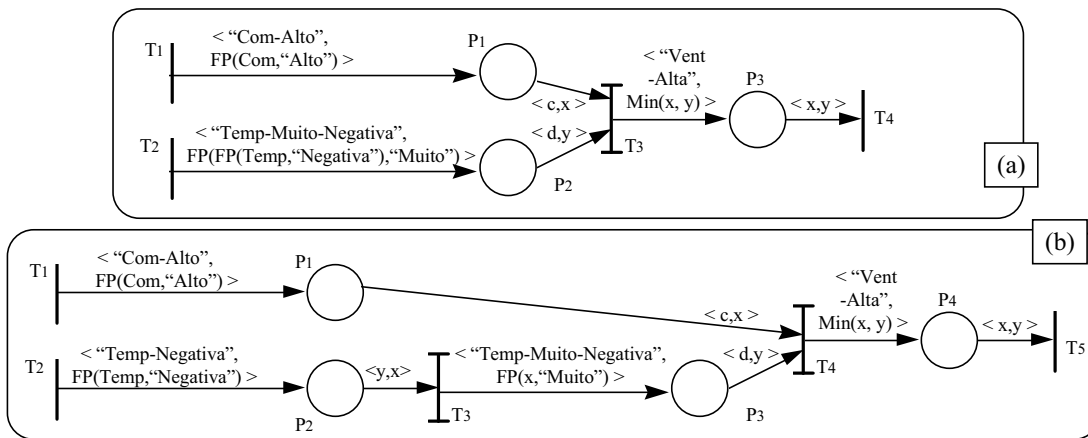


Figura 5-13 - Modelação de qualificadores:

a) num micropasso de computação; b) em dois micropassos.

A sua tradução em modelos RdP é obtida facilmente. Dois métodos básicos podem ser utilizados, ilustrados na Figura 5-13:

- o primeiro recorre a uma modelação utilizando um passo de computação, em que o valor de pertença é calculado por evocação recursiva à função de pertença FP;
- o segundo recorre a uma modelação utilizando computação (em cascata) do valor de pertença obtido através do modificador “Muito” aplicado ao nível de pertença associado ao valor linguístico “Negativa”.

As considerações produzidas na sub-seções anterior sobre a utilização de transições síncronas ou assíncronas também aqui são aplicáveis.

### 5.3.6 Descodificação difusa

Como analisado em capítulo anterior, a modelação de saídas nas RdP-R, como característica intrinsecamente não-autónoma, é realizada com base na marcação actual do modelo (tendo os autómatos de Moore como referência) ou na sua dinâmica (tendo os autómatos de Mealy como referência).

No caso particular de interesse nesta sub-secção, a determinação dos valores para sinais de saída associados a variáveis linguísticas é realizada considerando a marcação actual do modelo de acordo com o seguinte:

- considerando as marcas resultantes do processo de inferência cujos atributos linguísticos se encontram relacionados com a variável em causa;
- seleccionando um método de descodificação difusa adequado; de entre os porventura mais utilizados, refiram-se os seguintes métodos [Jager, 95]:
  - centro de gravidade;
  - centro de área (muitas vezes confundido com o centro de gravidade);
  - métodos indexados, que utilizam um nível mínimo de pertença na determinação do centro de gravidade ou de área;
  - média dos máximos.

Deste modo, a determinação dos valores dos sinais de saída caracterizados de forma difusa é realizada através da computação dos diferentes valores linguísticos e níveis de pertença associados, partindo do conhecimento do estado da RdP-R (marcação actual).

### 5.3.7 Representação alternativa de controladores difusos

Interessa referir as propostas de estruturas alternativas à de Mamdani-Assilian, nomeadamente a proposta em [Takagi & Sugeno, 83] [Takagi & Sugeno, 85].

O modelo de Takagi-Sugeno utiliza a caracterização difusa das entradas para definir os antecedentes das regras (de modo semelhante ao modelo de Mamdani-Assilian) conjugada com uma atitude de modelação das saídas através de funções de transferência (na linha dos modelos lineares por troços). Na proposta inicial, as funções utilizadas são da forma  $y_i = a_i x + b_i$ . O caso particular em que a função

é constante ( $y_i = b_i$ ), normalmente referido como o modelo de Sugeno, representa a intersecção entre os dois modelos referidos.

Os procedimentos de tradução apresentados nas sub-secções anteriores são válidos também para o modelo de Takagi-Sugeno, sendo as conclusões das várias regras modeladas por lugares (contendo uma marca colorida); a determinação dos valores das variáveis de saída é feita por computação de valores (mais uma vez), baseada nessas marcas e na informação das funções a utilizar.

### 5.3.8 Sobre a execução

A utilização de lógica difusa, presente nos controladores difusos, bem como em sistemas periciais difusos, tem encontrado um amplo leque de suportes à sua implementação. De um modo geral, são aplicáveis as dicotomias normalmente presentes no desenvolvimento de sistemas electrónicos digitais:

- software versus hardware, e
- arquitectura de aplicação geral versus arquitectura específica.

A selecção do modo de implementação dependerá, fundamentalmente, do desempenho e dos custos do sistema pretendidos (nos custos referidos incluem-se os custos de desenvolvimento e de produção). Como taxonomia geral, podem ser identificadas quatro classes de soluções [Costa et al., 95]:

- soluções (em software ou hardware) com componentes de aplicação geral;
- processadores de aplicação geral com instruções para computações especializadas;
- co-processadores difusos dedicados;
- “ASICs<sup>34</sup>” difusos com capacidade para operação autónoma.

Em [Costa et al., 95] as várias soluções apresentadas são detalhadamente caracterizadas. Na Figura 5-14 caracterizam-se as soluções típicas identificadas em termos da complexidade e tempo de resposta pretendido.

---

<sup>34</sup> ASIC - Application Specific Integrated Circuit, ou em Português, CIAE - Circuito Integrado de Aplicação Específica.

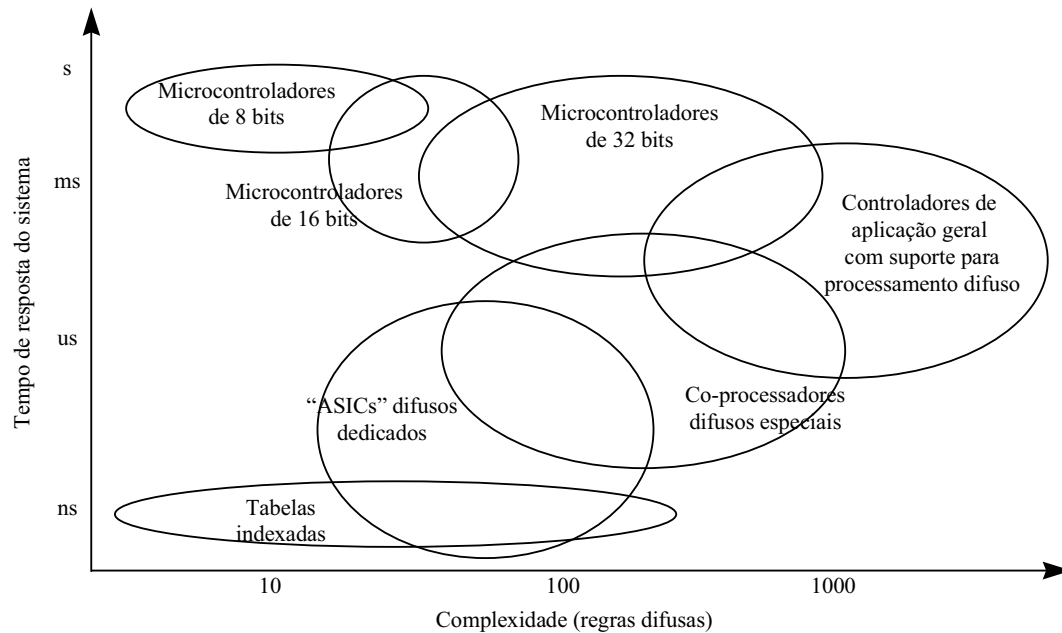


Figura 5-14 - Alternativas comuns para implementação em hardware (adaptado de [Costa et al., 95]).

Em particular, os métodos mais comuns para a implementação da fase de codificação difusa podem ser classificados em dois grupos:

- os que se baseiam na utilização de estruturas em memória, em que as funções de pertinência são avaliadas previamente e os resultados da sua digitalização armazenados em memória; recorrem à utilização de tabelas indexadas ou LUTs (“look-up tables”), em que os valores actuais dos sinais de entrada fornecem os índices para as tabelas; neste caso, o desempenho do sistema é independente do tipo de função de pertinência utilizado; como principal desvantagem, aponte-se a necessidade de recursos de memória eventualmente elevados; como exemplo, considere-se que o universo de discurso é representado por  $2^{10}$  valores, cada um dos quais codificado com 8 bits: será necessário utilizar uma memória com 1Kbyte e um acesso à memória é suficiente para a leitura de um valor de pertinência;
- os que se baseiam na avaliação da função de pertinência em tempo de execução; neste caso, apenas se necessita de armazenar os parâmetros característicos da função de pertinência especificada; a forma da função de pertinência utilizada é determinante no desempenho da implementação; como exemplo, considere-se a utilização de uma função triangular: será necessário armazenar apenas os três pontos notáveis da função e realizar algumas comparações e operações aritméticas para obter o valor de pertinência.

Do ponto de vista da execução do modelo RdP-R obtido, os métodos aplicáveis podem ser classificados em dois grupos [Gomes & Steiger-Garção, 95b] [Gomes & Steiger-Garção, 95c] (à semelhança do que se passa com qualquer modelo RdP-R):

- os que se baseiam na implementação directa da estrutura, no sentido dos jogadores-de-marcas, resultando na execução paralela da especificação; particularmente interessantes para implementação de sistemas com restrições severas de tempo-real;
- os que se baseiam na interpretação da estrutura, através da execução de uma representação do controlador, nomeadamente um espaço de estados compactado; dado que o número e localização das marcas no modelo RdP-R que representa o controlador difuso é constante, o seu estado pode ser representado pela parte dinâmica da sua marcação, isto é, os valores de pertença associados a todas as marcas (únicos que se alteram), como proposto em [Gomes & Steiger-Garção, 95a].

Ainda do ponto de vista da modelação das regras difusas e das suas implicações em termos da execução do modelo RdP-R associado a um controlador difuso, dado que a execução das RdP-R é realizada com base no conceito de ciclo de tratamento e de passo de análise, duas atitudes básicas podem ser adoptadas:

- a primeira, utilizando transições síncronas para modelar as várias regras difusas, permite a execução do modelo num único passo de análise; corresponde à computação instantânea do fluxo de dados resultante do disparo das transições responsáveis pela codificação difusa das variáveis de entrada;
- a segunda, utilizando transições assíncronas para modelar as várias regras difusas, possibilita a execução passo-a-passo do modelo, permitindo a propagação do fluxo de dados resultante da aquisição de nova informação a um ritmo de um nível de regras por passo de análise.

De realçar que ambas as atitudes fornecem resultados semelhantes, garantindo determinismo.

Um comentário final em relação às questões de verificação das propriedades do modelo. As técnicas de tradução propostas nas sub-secções anteriores impõem que as marcas que nascem nas transições-fonte (resultado da codificação difusa) percorrem o modelo da esquerda para a direita, vendo os seus valores de pertença combinados com os de outras marcas e sendo finalmente retiradas pelas transições à direita.

Dado que a utilização de transições-fonte corresponde, do ponto de vista do modelo abstracto, à existência de lugares não-limitados, logo não realizáveis, importa condicionar este comportamento. Para tal, no caso de utilização de transições síncronas para modelar as regras, deve-se considerar a existência de eventos externos associados às transições responsáveis pela codificação difusa que permitam o seu disparo apenas no início de um ciclo de execução do controlador.

Em qualquer das situações de modelação, para permitir a verificação do modelo (evitando a utilização de transições “desequilibradas”), é possível utilizar um modelo comportamentalmente semelhante ao apresentado. Para a obtenção desse modelo considere-se:

- a existência de uma única transição síncrona responsável pela fase de codificação difusa, isto é, reunindo numa única transição síncrona todas as transições anteriormente associadas a cada uma das variáveis linguísticas de entrada;
- a existência de uma única transição assíncrona responsável por retirar as marcas presentes nos lugares associados às conclusões;
- a criação de um lugar, inicialmente marcado com uma marca de baixo-nível, a considerar como lugar de entrada da transição síncrona referida no penúltimo parágrafo e lugar de saída da transição assíncrona referida no parágrafo anterior.

A Figura 5-15 ilustra o processo, quando aplicado ao exemplo da Figura 5-12.

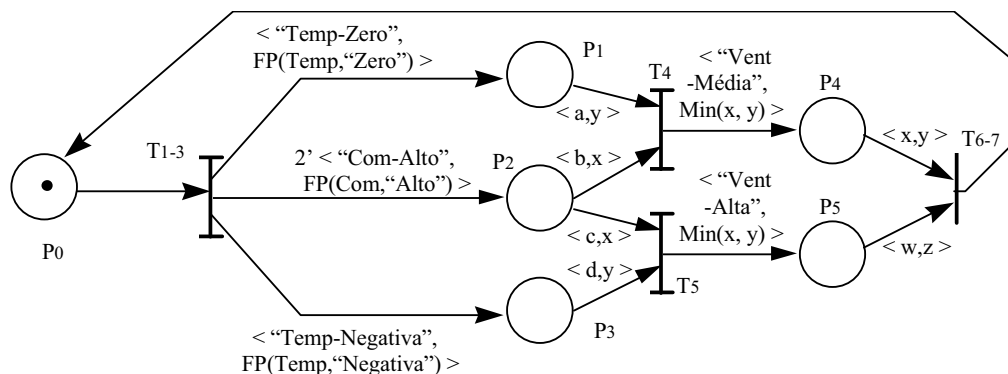


Figura 5-15 - Modelo comportamentalmente equivalente ao da Figura 5-12.

### 5.3.9 Organização hierárquica de controladores

Nas aplicações em que seja necessário utilizar um elevado número de variáveis linguísticas de entrada, é de interesse poder considerar os conceitos de:

- controladores difusos hierárquicos, e de
- diferentes granularidades em relação às variáveis de entrada.

O conceito de controlador difuso hierárquico advem da utilização de diferentes níveis de granularidade dos valores linguísticos incluídos nas regras [Pedrycz, 93]. A ideia de ampliação (“*zooming*”) corresponde, porventura, à solução mais simples que permite a implementação de

diferentes granularidades e está associada à atitude comum de abordagem de problemas, iniciando-se o processo por considerar condições mais gerais, evoluindo, posteriormente, para condições com maior detalhe. O ganho potencialmente obtido, em termos de desempenho do controlador, pode ser significativo, devido à redução do número de regras a avaliar.

A sua tradução em modelos RdP-R é trivial, como se ilustrará seguidamente e baseia-se na fusão de transições dos modelos RdP-R associados a controladores de níveis hierarquicamente dependentes, de modo a permitir a ligação em cascata das sub-redes associadas. A utilização de vários níveis (maiores que dois) de granularidade associados a diferentes conjuntos de regras difusas hierarquicamente dependentes é modelável de modo semelhante ao indicado para o caso de dois níveis.

Exemplo 5.5: Considere-se um conjunto de regras difusas, denominado por controlador difuso C1, em que se utilizam partições difusas das variáveis de entrada com “poucos” valores linguísticos possíveis. Dependendo das conclusões associadas a este conjunto de regras, (no exemplo a utilizar, do produto dos níveis de pertença das suas conclusões), optar-se-á pela execução de um de dois possíveis controladores, denominados por C21 e C22; os controladores deste segundo nível utilizam partições difusas das variáveis de entrada mais detalhadas. A Figura 5-16 ilustra a situação referida. A activação de um (e só um) dos controladores C21 e C22 é garantida através da utilização de guardas adequadas (mutuamente exclusivas) associadas às transições de saída do controlador C1 (“ $x*y>a$ ” e “ $x*y\leq a$ ”, em que  $a$  representa um nível *adequado* de comparação).

No exemplo apresentado, à especificação hierárquica inicial (em termos das variáveis linguísticas utilizadas) faz-se corresponder um modelo RdP-R plano em que, de uma forma geral, algumas sub-redes são activadas como resultado da execução de outras sub-redes.

Implementações alternativas para o conceito de controlador difuso hierárquico recorrem às capacidades de estruturação hierárquica das RdP-RH. Duas hipóteses são de mencionar:

- a primeira em que se consideram as dependências hierárquicas, não em termos gráficos como realizado no exemplo apresentado, mas em termos da execução, como o permitido através da utilização de metanós e meta-redes;
- a segunda em que, para além do recurso a metanós, se considera possível a activação concomitante de vários controladores hierarquicamente dependentes, com graus de ponderação determinados por características da super-rede associada (nomeadamente o valor de pertença associado a determinadas marcas, isto é, valores linguísticos).

No final da próxima secção, estas capacidades das RdP-RH serão ilustradas.

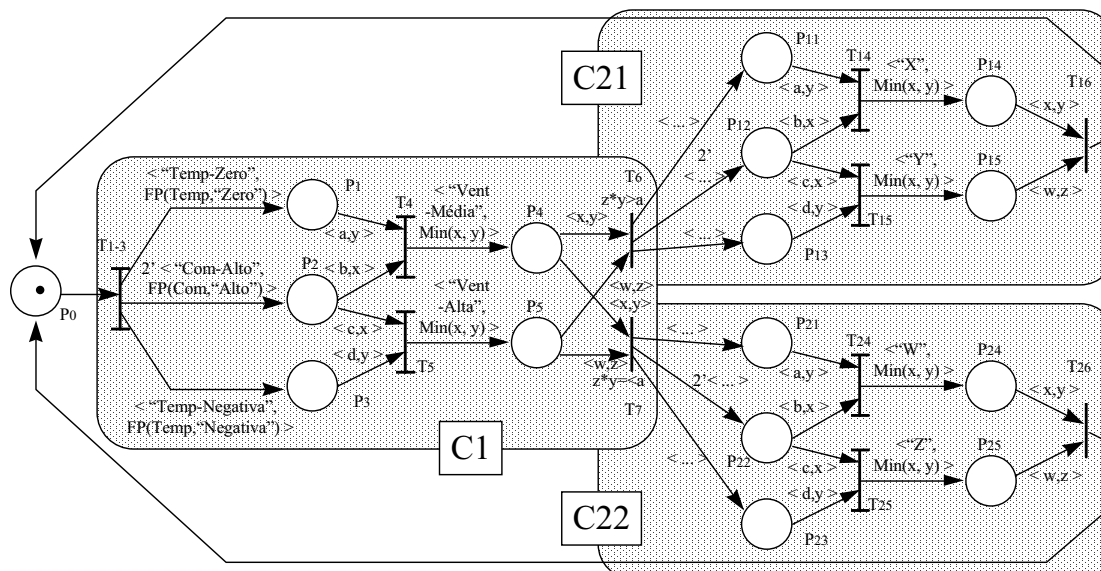


Figura 5-16 - Junção de controladores hierarquicamente organizados.

### 5.4 Aplicação a controladores programáveis

Nas linguagens normalmente utilizadas para a programação de controladores industriais, comumente referidos por PLCs (de “Programmable Logical Controller”) ou autómatos programáveis (por adaptação do termo francês), incluem-se a linguagem de contactos (“ladder diagrams”), o Grafcet e linguagens lógicas [Silva, 89].

De realçar que o Grafcet é o resultado de actividades de normalização, envolvendo industriais e académicos franceses, tendo recebido uma forte contribuição das RdP e tendo conduzido, posteriormente, à norma internacional IEC-848, produzida no seio do Comissão Electrotécnica Internacional [Silva, 89] [David & Alla, 92]. Os esforços internacionais de criação de normas actuais aplicáveis à programação de controladores têm-se mantido, sendo a norma IEC-1131, um exemplo relevante dessas actividades.

A utilização de extensões às linguagens de programação “comuns” (normalmente C) ou de outras baseadas em blocos lógicos são, no entanto, comuns. Como exemplo com interesse particular no mercado português, refira-se a linguagem ABL (Advanced Boolean Language) concebida pela Patrónica para a sua linha de controladores, com capacidades de especificação do controlo através de um circuito lógico [Patrónica, 93] [Prata & Nunes, 93].

Neste cenário, em que os métodos de especificação de controlo, em sintonia com as características dos controladores utilizados, são relativamente pouco sofisticados e na generalidade dos casos estritamente ligados a uma linha de controladores/fabricante, é de elevado interesse, como referido em

capítulos anteriores, possibilitar a integração de especificações produzidas tendo como referência outros paradigmas de modelação de sistemas.

Previamente os aspectos demonstrativos, será apresentado um exemplo de aplicação à monitoração e controlo de edifícios, como exemplo de controlo de arquitecturas contendo sensores e actuadores.

Considere-se, pois, a título de exemplo, um caso típico do controlo de um sistema de ar condicionado. Sem prejuízo do exposto, considera-se que o controlador utilizado executa paralelamente outras especificações de controlo, garantindo-se os níveis de desempenho exigidos pelos vários processos controlados.

O utilizador pode especificar um modo de funcionamento e a temperatura pretendida. O modo de funcionamento pode ser “Desligado”, “Alto”, “Médio” e “Baixo” e determina a velocidade da ventoinha; a temperatura pretendida pode estar entre 10 e 30 graus Centígrados. Estão disponíveis sensores de temperatura, um instalado no interior do espaço sob controlo e outros no exterior, bem como detectores binários que permitem concluir sobre o estado das portas e janelas (aberto/fechado). Para simplificar a exposição, considerem-se apenas duas saídas a controlar, associadas à ventoinha e aquecimento (refrigeração é ignorada).

Partindo de uma especificação simples em que sempre que uma porta ou janela se encontram abertas a ventoinha pára e em que, caso contrário, o modo de funcionamento pretendido é seguido, o sistema pode ser facilmente descrito por uma máquina de estados como a que se caracteriza na Figura 5-17, em que o estado da ventoinha é determinado directamente pelo estado activo.

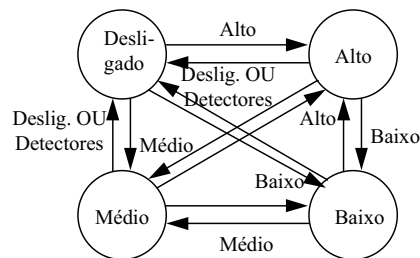


Figura 5-17 - Especificação inicial de controlo do sistema de ar condicionado.

Procedimento igualmente simples (embora pouco eficiente) permitiria determinar o aquecimento necessário, tendo em conta a diferença entre as temperaturas medida e pretendida.

No entanto, se fosse pretendido implementar outra estratégia de controlo, integrando, por exemplo, a gestão da energia eléctrica consumida e sua articulação com o regime de funcionamento pretendido para a instalação global, a obtenção de um modelo de controlo associado poderia ser tarefa complexa. Considere-se, a título de exemplo, o objectivo de manter o consumo de pico da instalação global

abaixo dos níveis contratados com a empresa fornecedora de energia eléctrica, assegurando, em situações de sobrecarga, uma degradação “suportável” no funcionamento do sistema de ar condicionado.

São conhecidas aplicações de controlo difuso e neuronal nestas situações, permitindo a obtenção de curvas de variação de temperatura mais suaves e evitando oscilações em torno do valor pretendido (e consequentes consumos de energia adicionais).

Na exposição que se segue, ilustra-se, numa primeira fase, a introdução de controlo difuso, sendo, numa segunda fase, ilustrada a utilização de raciocínio aproximado, de modo a substituir a máquina de estado apresentada na Figura 5-17 por um conjunto de regras com imprecisão.

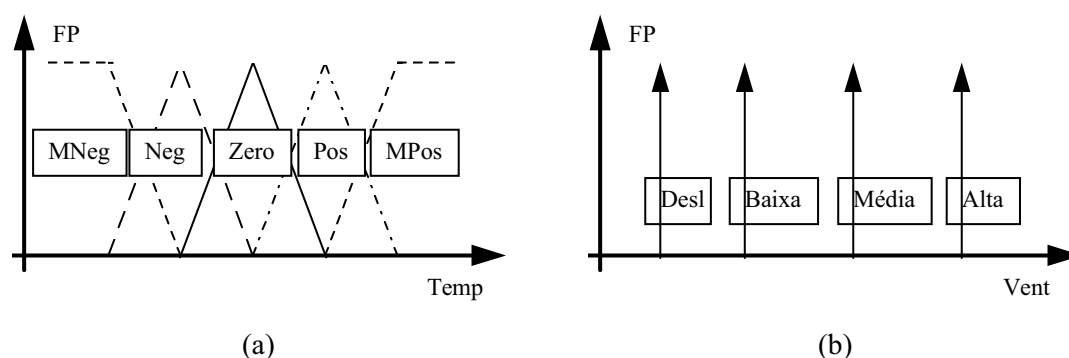


Figura 5-18 - Partição difusa das variáveis Temperatura (a) e Ventoíinha (b).

Assim, considere-se que se substitui a actuação directa das saídas dependente do modo de funcionamento por um controlador difuso. São utilizadas duas variáveis de entrada (*Comando* e *Temperatura*, referidas nas figuras por *Com* e *Temp*, respectivamente) e uma de saída (*Ventoíinha*, referida nas figuras por *Vent*). As partições difusas associadas apresentam-se na Figura 5-9 e na Figura 5-18. Como usual nas aplicações de controlo em que o suporte directo de implementação é hardware, a variável linguística de saída é caracterizada através de um conjunto de valores singulares (“*singletons*”). A variável de entrada *Temperatura* é o resultado da diferença dos sinais de entrada “*Temperatura pretendida*” e “*Temperatura medida*”.

Considere-se, a título ilustrativo, o conjunto de regras apresentadas no Exemplo 5.4, que se reproduzem seguidamente.

IF Temperatura IS Zero AND Comando IS Alto THEN Ventoíinha IS Média

IF Temperatura IS Negativa AND Comando IS Alto THEN Ventoíinha IS Alta

Modelos RdP-R resultantes da sua tradução foram apresentados na Figura 5-12 e na Figura 5-15. Para a sua implementação utilizar-se-á a linguagem ABL [Fatrónica, 93] e o controlador programável

PLC1B [Prata & Nunes, 93]. A linguagem ABL é utilizada como exemplo, não sendo considerada como redutora da aplicabilidade dos procedimentos referidos a outras famílias de controladores. A tradução das formas mais comuns, como linguagem de contactos em ABL, é uma tarefa que não levantará, na maior parte dos casos, problemas de grande complexidade.

Os modelos presentes na Figura 5-12 e na Figura 5-15 (comportamentalmente equivalentes) representam, basicamente, o fluxo de dados associado à aquisição e agregação de valores de pertença dos valores linguísticas de entrada. Essa é a razão para a utilização de transições síncronas, cujo disparo é imediato (recorde-se que a execução do modelo RdP-R é realizada tendo por base o conceito de passo de análise e ciclo de tratamento; o disparo das transições síncronas, desde que habilitadas e prontas, realiza-se sempre, independentemente do instante de criação das marcas que as habilitam). A característica de “processamento de dados” instantâneo, realizado pelas transições síncronas utilizadas, evita que os lugares intermédios sejam sincronizados (que sejam implementados por elementos de memória, tipo biestáveis). Desta forma, a implementação de hardware dedicado à sua execução, um jogador-de-marcas específico, mais não é que uma arquitectura não sincronizada de processamento do fluxo de dados modelado.

Devido às características das transições síncronas, evidenciadas no parágrafo anterior, a determinação da parte dinâmica da marcação (referida, em subsecção anterior, como método de implementação alternativo) é coincidente com a implementação referida para o processamento do fluxo de dados.

Deste modo, o estado do modelo RdP-R da Figura 5-15, representado pela parte dinâmica da sua marcação, isto é, pelos valores de pertença das várias marcas utilizadas, será determinado com base na tradução em ABL dos cálculos que afectam esses valores de pertença.

Na Figura 5-19 apresenta-se a implementação parcial da fase de codificação difusa referente à variável Temperatura, em que as funções de pertença são realizadas através de tabelas indexadas - LUTs (“look-up tables”) de reduzida/mediana dimensão, representadas pelos blocos ROMA (de ROM Array). Os valores fornecidos à saída representam o valor da função de pertença dos vários valores linguísticos. A representação do valor de pertença, normalmente pertencente ao intervalo  $[0,1]$ , é feita através de inteiros; no caso implementado utilizou-se o intervalo  $[0,63]$ , com 6 bits na representação, sendo os erros obtidos não significativos. A utilização de qualificadores nas proposições utilizadas, como “Muito Zero”, é facilmente implementada através de LUTs ligadas em cascata, como se apresenta na parte superior direita da Figura 5-19.

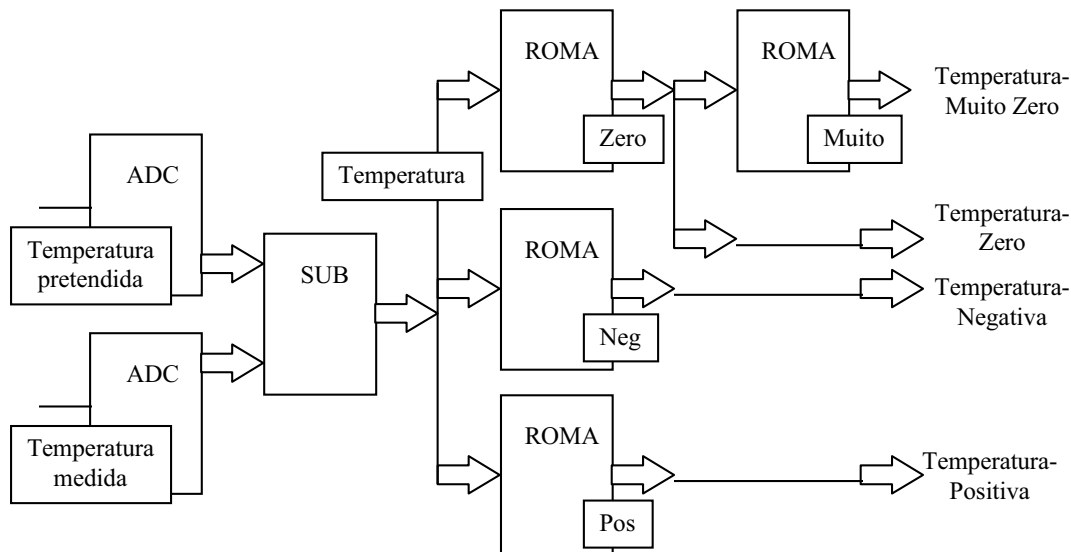


Figura 5-19 - Código ABL para fase de codificação difusa.

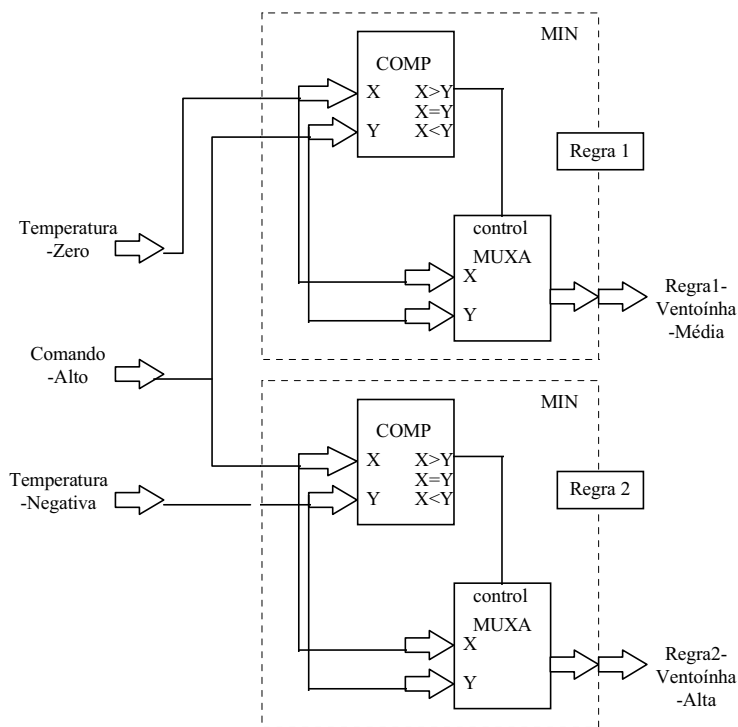


Figura 5-20 - Código ABL para fase de inferência difusa.

A Figura 5-20 apresenta a implementação da fase de inferência difusa, em que o operador MIN foi adoptado para realizar a composição AND dos valores de pertinência associados às várias proposições difusas.

A Figura 5-21 apresenta uma implementação para a fase de descodificação difusa, em que se realiza a agregação ALSO das várias regras pelo operador SUM e se utilizou o centro de gravidade como método de descodificação difusa. Os vários blocos utilizados foram implementados como módulos ABL. Deste modo, a expressão que permite calcular o valor da variável de saída é a seguinte:

$$Variavel = \frac{\sum_i mvi * fi}{\sum_i mvi}$$

em que:

- $mv_i$  transporta o valor de pertença resultante da composição AND dos antecedentes da regra  $i$ , e
- $f_i$  é o valor numérico associado ao valor linguístico singular (“*singleton*”) utilizado na regra  $i$ .

Os resultados obtidos revelam que, devido à aritmética inteira utilizada, torna-se necessário prestar especial atenção à geração e propagação de erros.

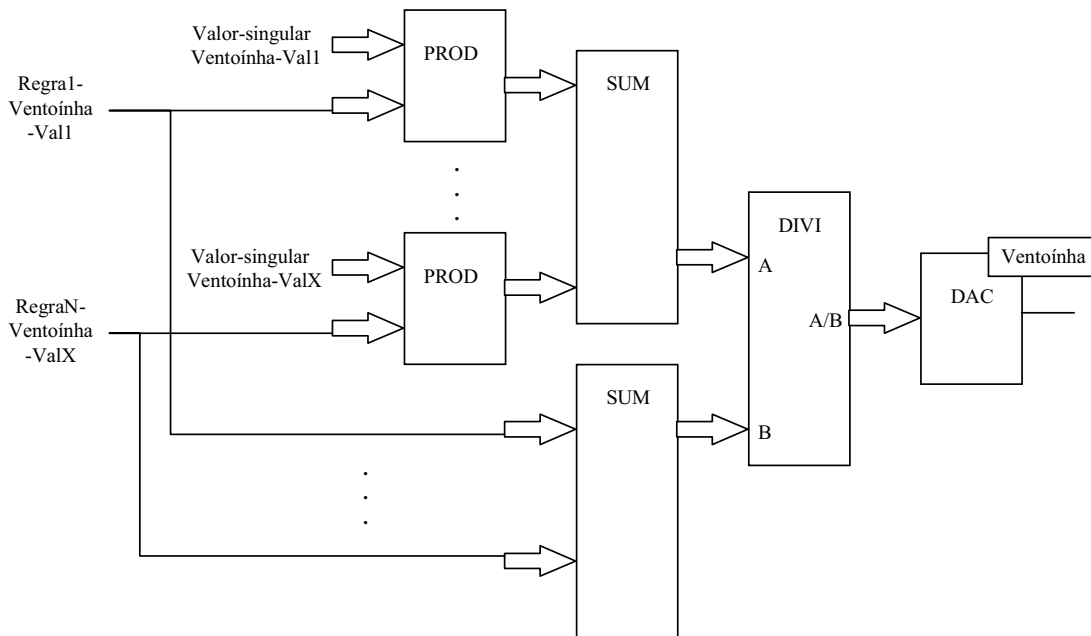


Figura 5-21 - Código ABL para fase de descodificação difusa.

Finalmente, no sentido de ilustrar a utilização de raciocínio difuso, considere-se a utilização de um sistema de regras, permitindo integrar na especificação estratégias de controlo não-convencional. Como referido, a utilização de regras com imprecisão e de regras difusas recebem o mesmo tipo de procedimentos de tradução em RdP-R, pelo o que se exporá é aplicável em ambas as situações.

Desta forma, considere-se que a “folga” de potência disponível entre o valor acordado com a empresa fornecedora de energia eléctrica e a potência instantânea efectivamente consumida é fornecida pelos sistemas instalados na planta do edifício. O objectivo do controlo é conseguir que, à custa de uma eventual degradação no funcionamento do sistema de ar condicionado (mantendo, no entanto, os níveis mínimos de conforto), seja possível reduzir consumos no sistema de ar condicionado de modo que globalmente não se ultrapasse o escalão contratado.

A substituição da máquina de estados da Figura 5-17 por um sistema de regras torna-se, assim, interessante.

Considerando que o sistema de tarifação eléctrica fornece valores para a “folga” de potência disponível, representada através da variável linguística “*Energia*”, sendo produzidos, através da sua codificação difusa, um conjunto de factos do tipo:

Energia IS Pouca CF=x

Energia IS Média CF=y

Energia IS Muita CF=z

O seguinte sub-conjunto de regras permite ilustrar o método de determinação do estado de funcionamento do sistema:

IF Energia IS Pouca AND Comando IS Alto THEN Estado IS Médio

IF Energia IS Pouca AND Comando IS Médio THEN Estado IS Baixo

IF Energia IS Pouca AND Comando IS Baixo THEN Estado IS Baixo

em que a partição difusa da variável Comando foi atrás apresentada (Figura 5-9).

A representação (parcial) através de RdP-R deste conjunto de regras apresenta-se na Figura 5-22, explicitando-se a evolução do atributo “nível de pertença” associado às marcas (omite-se a transição responsável pela remoção das marcas dos lugares  $P_7$  e  $P_8$ ). A sua tradução em ABL (porque semelhante à apresentada anteriormente) é omitida.

A combinação de várias marcas associadas a um mesmo valor linguístico (nomeadamente as marcas presentes nos lugares  $P_7$  e  $P_8$ ) são explicitamente agregadas numa única marca através do disparo da transição  $T_6$ .

Obtêm-se, assim, um conjunto de lugares representativos dos estados “Alto”, “Baixo” e “Médio” representados na Figura 5-17. Concretamente, o lugar  $P_8$  representa o antigo estado “Médio” e o lugar  $P_7$  o antigo estado “Baixo”. No entanto, neste caso, os vários estados vão estar simultânea e parcialmente activos.

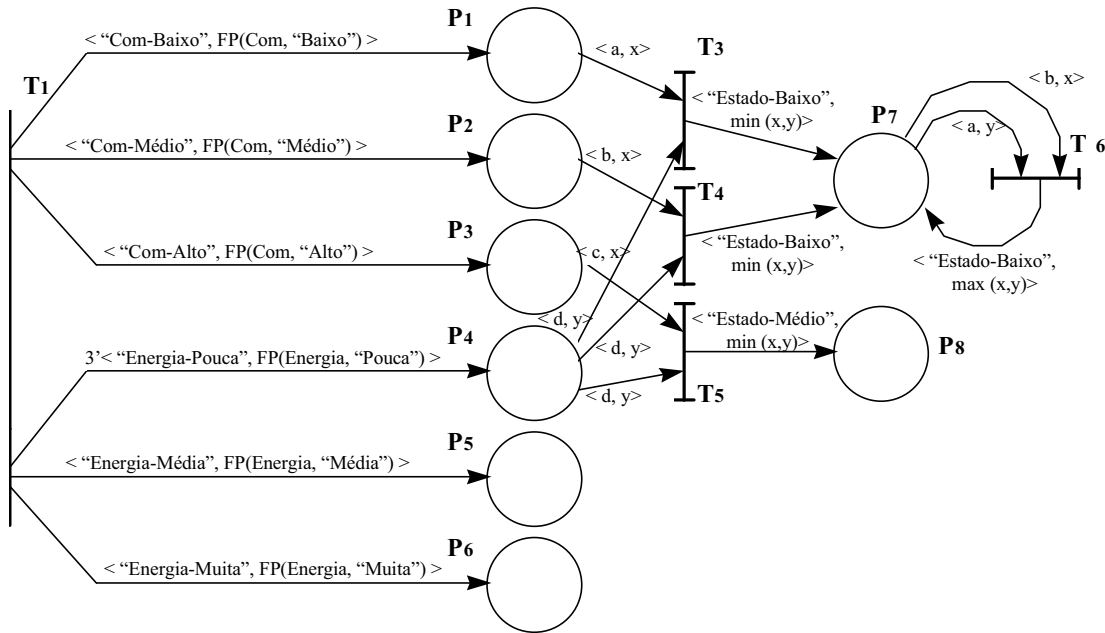


Figura 5-22 - Modelo RdP-R de regras difusas associadas a nível de topo.

Várias atitudes de projecto podem ser adoptadas neste ponto; como exemplos, considerem-se as seguintes:

- proceder à descodificação difusa da variável “Estado” e utilizá-la como entrada “Comando” do controlador difuso já apresentado; desta forma, existiam dois controladores difusos organizados hierarquicamente em que a ligação entre os dois é realizada totalmente através de variáveis de entrada e saída;
- considerar que os lugares  $P_7$  e  $P_8$  são metalugares, aos quais estarão associados meta-redes contendo a especificação de controlo adequado a cada um dos estados; a fase de descodificação difusa é realizada em relação ao conjunto de sub-modelos de RdP-R em execução, em cada instante; a herança de características da RdP-R do nível de topo (representado na Figura 5-22) para as meta-redes dependentes é realizada de acordo com as características das RdP-RH, permitindo, nomeadamente, a comunicação à meta-rede dos valores de pertença actuais associados aos diferentes valores linguísticos.

Deste modo, é possível observar a activação simultânea de vários controladores difusos, com contribuições ponderadas, ao nível das acções, pelos níveis de pertença associados aos diferentes valores linguísticos.

## **5.5 Retomando a resolução automática de conflitos**

Como detalhadamente apresentado em capítulo anterior, a resolução automática de conflitos pode ser adequadamente resolvida através da utilização de sub-redes que se comportem como árbitros. As possibilidades de implementar através de RdP-R mecanismos de inferência e de os embeber nesses árbitros especializados em resolução de conflitos, dispõe, deste modo, de um enorme potencial de aplicação em situações da vida real.

Como exemplo de uma área de aplicação em que heurísticas “simplistas” têm um interesse real muito limitado e onde a utilização de mecanismos de inferência difusa podem vir a ter um impacto muito significativo, refira-se a área de controlo integrado de grupos de elevadores. É uma área em que se prevê a realização de protótipos a breve prazo (no âmbito do projecto CYTED-SIntEd) e onde a especificação de critérios para definir disciplinas de atendimento de chamadas de elevadores reveste-se de elevada complexidade.



## 6. Edifícios inteligentes: um conceito potencialmente mobilizador

### *Sumário do capítulo*

A área de aplicação dos edifícios inteligentes desde cedo foi identificada, nos trabalhos de que esta tese é um resultado, como uma das áreas principais de aplicação. Devido à diversidade das subáreas presentes, constituiu-se como uma fonte de situações a modelar para o teste e validação das características propostas para as novas classes de RdP. Os níveis de abstracção a utilizar ou os tipos de sistemas a modelar podem ser extremamente diversos. De entre eles consideraram-se, como exemplo, os sistemas distribuídos vocacionados para a monitoração e controlo de edifícios. Os seus vários aspectos podem ser objecto de modelação através de RdP-R, desde os de nível de abstracção mais elevado (como as arquitecturas cliente-servidor e os protocolos de comunicação associados), até aos níveis de actuação e de aquisição de dados sensoriais (como os controladores específicos). Essa a razão para que a maioria dos exemplos apresentados, ao longo dos capítulos anteriores, seja relativa a esta área de aplicação.

O capítulo inicia-se (na sua primeira secção) com um levantamento do estado da arte da área, do ponto de vista da contribuição das tecnologias da informação, enquanto que na segunda secção se enfatiza o papel que o tema pode assumir no contexto da indústria e dos serviços, bem como no meio académico.

Nas duas secções seguintes, após a caracterização do enquadramento das propostas e protótipos a apresentar em secções posteriores, caracterizam-se os ocupantes dos edifícios, agrupando-os em três grupos: os utilizadores das infraestruturas, os administradores de recursos e os domots. Os domots (cujo nome é resultado da concatenação das palavras “domus” e “robot”) são entidades computacionais com capacidade de interacção com os ocupantes humanos e com a infraestrutura predial.

Nas três secções seguintes apresenta-se o conjunto de propostas e protótipos desenvolvidos, tendo como objectivo a monitoração e controlo de edifícios através de arquitecturas distribuídas. Serão descritas as arquitecturas de referência propostas, quer para a interligação física de sistemas, quer para a interligação dos processos computacionais e de controlo, bem como os protótipos mais relevantes desenvolvidos.

Na última secção abordar-se-ão situações e necessidades de modelação específicas presentes nesta área de aplicação em que se prevê a utilização no futuro próximo das RdP-R e as RdP-RH.

Deste modo, este capítulo é composto pelas seguintes secções:

*6.1 Introdução*

*6.2 Um conceito potencialmente mobilizador para a indústria e para os serviços*

*6.3 Enquadramento das propostas a apresentar*

*6.4 A necessidade de novos ocupantes: os domots*

*6.5 Perspectivas de modelação*

*6.6 Ambientes de desenvolvimento e exploração em edifícios*

*6.7 Arquitecturas de referência*

*6.8 Análise de casos*

## 6. Edifícios inteligentes: um conceito potencialmente mobilizador

### 6.1 Introdução

Durante a década de 80, apreensões no tocante a algumas utilizações de energia, associadas a estratégias tendentes a reduzir custos com a utilização e manutenção de edifícios, conduziram ao conceito de “edifício inteligente”<sup>35</sup> (EI). Este conceito tem vindo a ser interpretado de variadas formas, distorcido e redefinido. Outras referências, algumas das quais entretando caídas em desuso, como edifícios espertos, de elevada tecnologia, integrados ou de tecnologia avançada<sup>36</sup>, têm vindo a ser utilizadas. Mais recentemente, o termo “Domótica” tem vindo a ser aceite como identificador desta área de aplicação, considerando apenas os aspectos relativos às tecnologias de informação.

Sem dúvida que é uma área de aplicação multidisciplinar, onde diferentes áreas do conhecimento podem contribuir de forma complementar, no sentido do benefício quer dos ocupantes ou utilizadores dos edifícios, como também dos proprietários ou de quem proceda à sua exploração. Benefícios mais difíceis de quantificar, quer em termos económicos, quer em termos ecológicos, podem ser associados à natural poupança energética. De entre os domínios do conhecimento com contribuições relevantes nesta área de aplicação destacam-se a arquitectura, as engenharias civil, electrónica e de computadores, nomeadamente de sistemas, controlo e telecomunicações.

Várias análises sobre a evolução do conceito de edifícios inteligentes têm sido apresentadas na bibliografia da área (que para além de escassa, não está sistematizada). Uma perspectiva histórica, particularmente interessante, é apresentada em [DEGW & Teknibank, 95] e retomada em [DEGW, Arup & Northcroft, 96]. Aí, o desenvolvimento do conceito de edifício inteligente é caracterizado através de três fases:

---

<sup>35</sup> “Edifício inteligente” ou EI, tradução de “intelligent building” ou IB.

<sup>36</sup> “Edifícios espertos, de elevada tecnologia, integrados ou de tecnologia avançada”, traduções para “smart, high-tech, integrated or advanced technology buildings”.

- a primeira, associada a “edifícios automatizados<sup>37</sup>”, decorre durante o período de 1981 a 1985, em que o conceito de edifício inteligente é encarado como uma colecção de tecnologias inovadoras, permitindo a gestão do edifício, o escritório electrónico e comunicações;
- a segunda, associada a “edifícios responsáveis<sup>38</sup>”, decorre durante o período de 1986 a 1991, em que o edifício inteligente é encarado como uma colecção de tecnologias, com capacidade de responder às mudanças organizativas ao longo do tempo;
- a terceira e última, associada a “edifícios efectivos<sup>39</sup>”, decorre após 1992, em que o edifício inteligente fornece um ambiente responsável, efectivo e sustentado, no qual a organização pode alcançar os seus objectivos.

Sendo, potencialmente, uma área de aplicação com enorme impacto quer na vida das pessoas, quer no funcionamento das empresas e instituições, com potencialidades de influenciar mercados com enormes volumes de vendas, como, por exemplo, do imobiliário e da electrónica de consumo, cedo se verificou o aparecimento *ad-hoc* de grupos de interesse, tendencialmente estruturantes e geradores de normas e recomendações, condicionantes da actuação das empresas e do aparecimento de novos produtos.

De referir o Intelligent Buildings Institute (IBI), com sede em Washington, Estados Unidos da América, entretanto sem actividade significativa, para quem “um edifício inteligente é o que fornece um ambiente produtivo e com custos competitivos através da optimização dos seus quatro elementos básicos: Estruturas, Sistemas, Serviços e Gestão, e as interrelações entre eles... A característica, a única característica”, do ponto de vista do IBI, “que todos os edifícios inteligentes necessitam de ter em comum é a estrutura projectada para acomodar a **alteração** de modo conveniente e com custos efectivos” [Becker, 95].

Por outro lado, a definição adoptada pelo European Intelligent Building Group (EIBG) considera que “um edifício inteligente cria um ambiente que permite às organizações atingir os seus objectivos naturais e maximiza a eficiência dos seus ocupantes enquanto ao mesmo tempo permite uma gestão eficiente dos recursos com um mínimo de custos em termos de ocupação humana” [Becker, 95].

Em [Shemie, 97], um “Edifício Inteligente deve fornecer um ambiente confortável e adaptável para o utilizador, flexível e com custos eficazes para a organização e atractivo e capaz para o proprietário;

---

<sup>37</sup> “Edifícios automatizados”, tradução de “automated buildings”.

<sup>38</sup> “Edifícios responsáveis”, tradução de “responsive buildings”.

cada sistema irá interagir, de alguma forma, com todos os outros e todos os sistemas devem trabalhar em harmonia se se pretender alcançar benefícios plenos para o utilizador”.

As definições apresentadas têm como objectivo a criação de um ambiente eficiente e produtivo, através de custos mínimos. Nos já referidos trabalhos [DEGW & Teknibank, 95] e [DEGW, Arup & Northcroft, 96], produzidos com a colaboração do “European Intelligent Building Group”, são apresentados dois estudos de mercado de particular relevância relativos à Europa e ao Sudeste Asiático. Em [Lustig, 95] e [Lustig, 97], como resultado da realização de dois congressos em “Edifícios Inteligentes”, pode ser encontrado um conjunto relevante de propostas.

Do ponto de vista da classificação das funções do edifício inteligente várias têm sido as propostas apresentadas. Como referido em [Arkin & Paciuk, 95], em [McKinley, 88] propõe-se uma divisão em (ver Tabela 6-1):

- Sistemas,
- Serviços, e
- Gestão.

| <b>Funções</b>  | <b>Descrição</b>  |
|-----------------|---|
| <b>Sistemas</b> | AVAC (aquecimento, ventilação e ar condicionado), iluminação, energia eléctrica, elevadores, drenagem e abastecimento de água, controlo de acessos, segurança (fogo e vida), intrusão, telecomunicações e processamento de informação, etc... |
| <b>Serviços</b> | voz/video/dados, escritório electrónico, serviços (partilhados) a locatários, gestão de segurança, operações fora-de-horas, etc...  |
| <b>Gestão</b>   | incluindo limpeza, manutenção, treino, propriedade, arrendamento, tecnologia, etc...  |

Tabela 6-1 - Classificação das funções do edifício inteligente (segundo [McKinley, 88]).

Ainda segundo [Arkin & Paciuk, 95], uma classificação alternativa atribuída a Klein, proposta em 1988, enfatiza os serviços disponibilizados para os ocupantes, considerando quatro categorias:

<sup>39</sup> “Edifícios efectivos”, tradução de “effective buildings”.

- automação predial<sup>40</sup> (coincidindo com a categoria “sistemas” da classificação de [McKinley, 88]);
- serviços a locatários (combinando as categorias “serviços” e “gestão” de [McKinley, 88]);
- telecomunicações (consideradas como “sistemas” em [McKinley, 88]);
- escritório electrónico.

De particular interesse, refira-se a caracterização da evolução dos conceitos associados ao edifício inteligente, devida ao EIBG, resultando a Figura 6-1 como síntese [DEGW & Teknibank, 95], em que se identificam três estágios de integração associados ao mercado de edifícios inteligentes:

- sistemas de automação predial<sup>41</sup>, onde se integram os sistemas de segurança e controlo;
- sistemas de comunicação integrados, onde se integram as comunicações, escritório electrónico e gestão;
- edifício computacionalmente integrado<sup>42</sup>, onde os dois grupos de sistemas referidos se encontram integrados.

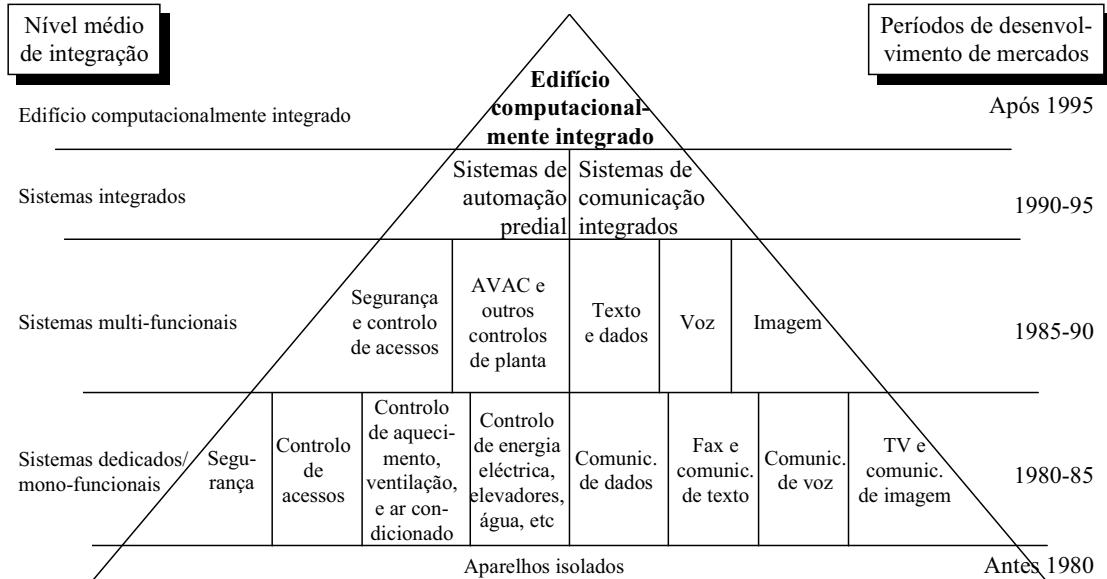


Figura 6-1 - A pirâmide do Edifício Inteligente (adaptado de [DEGW & Teknibank, 95])

<sup>40</sup> “Automação predial”, tradução de “building automation”.

<sup>41</sup> “Sistemas de automação predial”, tradução de “building automation systems”.

<sup>42</sup> “Edifício computacionalmente integrado”, tradução de “computer integrated building”.

É nítida e significativa a relação directa entre os períodos de desenvolvimento de edifícios inteligentes e o desenvolvimento das tecnologias de informação.

Interessante, por outro lado verificar, que a escala de tempo associada a esta pirâmide (que é extensivamente utilizada como referência em comunicações e relatórios públicos internacionais), foi recentemente “reajustada”, reflectindo uma forte “desaceleração” nas expectativas à volta dos edifícios inteligentes.

Em [Shemie, 97], onde se utiliza uma pirâmide em que não se considera o nível básico de “Aparelhos isolados”, é particularmente significativa a alteração em 10 anos na data prevista para alcançar o “edifício computacionalmente integrado”, sendo a escala de tempos reajustada da seguinte maneira:

|                         |   |      |
|-------------------------|---|------|
| Após 1995               | → | 2005 |
| 1990-95                 | → | 1996 |
| 1985-90                 | → | 1985 |
| 1980-85 e Antes de 1980 | → | 1970 |

A aplicação efectiva das tecnologias de informação é, reconhecidamente, necessária ao sucesso de qualquer instituição ou empresa. A sua integração no edifício inteligente de modo sustentado e efectivo é necessária à procecussão dos objectivos da organização.

Como consequência natural e directa das diversas contribuições recebidas, a definição de alguns novos conceitos, como “inteligência predial”<sup>43</sup>, tem sido realizada em diferentes formas, por vezes complementares, dependendo da comunidade em causa. Para o Intelligent Buildings Institute, a inteligência do edifício necessita de ser avaliada em função dos quatro elementos básicos já referidos, estrutura, sistemas, serviços e gestão [IBI, 87]. A abrangência desta definição implica que a inteligência do edifício necessita de começar a ser definida e preparada desde o início do projecto, mantendo-se durante a sua operação e exploração.

Em [DEGW & Teknibank, 95] consideram-se três níveis de soluções condicionantes da inteligência predial, entendidos como dispostos em camadas (ver Tabela 6-2):

- disponibilidade de estruturas eficientes<sup>44</sup>;
- aplicação das tecnologias de informação no sentido de reduzir custos e melhorar o desempenho;

---

<sup>43</sup> “Inteligência predial”, tradução de “building intelligence”.

<sup>44</sup> “Estruturas eficientes”, tradução de “effective building shell”.

- disponibilidade de tecnologias e serviços integrados.

| <b>Tipo de solução</b>                                      | <b>Caracterização</b>  |
|---|--|
| <b>Estruturas eficientes</b>                                | <ul style="list-style-type: none"> <li>• altura dos pisos</li> <li>• profundidade</li> <li>• configuração</li> <li>• capacidade de carga por piso</li> <li>• capacidade para comunicações</li> <li>• revestimento</li> </ul>   |
| <b>Aplicação das tecnologias da informação</b>              | <ul style="list-style-type: none"> <li>• edifício</li> <li>- controlo ambiental do edifício</li> <li>- controlo do edifício por parte dos ocupantes</li> <li>• espaços</li> <li>- gestão de alterações</li> <li>- monitoração de utilização</li> <li>- controlo de acessos</li> <li>• actividade</li> <li>- comunicações internas e externas</li> <li>- processamento e armazenamento de informação</li> <li>- apresentação de informação</li> </ul> |
| <b>Disponibilidade de tecnologias e serviços integrados</b> | <ul style="list-style-type: none"> <li>• CAFM (Computer Aided Facility Management)</li> <li>• LANs &amp; WANs (redes de dados locais e globais)</li> <li>• sistemas de automação predial</li> <li>• sistemas integrados de informação</li> <li>• serviços de consultadoria</li> </ul>  |

Tabela 6-2 - Tipos de solução característicos da inteligência predial  
(segundo [DEGW & Teknibank, 95]).

Do ponto de vista da aplicação das tecnologias de informação, e ainda segundo [DEGW & Teknibank, 95], três áreas de aplicação devem ser consideradas:

- em sistemas de automação predial, incluindo o controlo ambiental e permitindo o controlo personalizado por parte dos ocupantes dos sistemas de automação instalados;
- em sistemas de gestão de espaços, permitindo gerir facilmente alterações, monitorar utilizações, controlar acessos e minorar custos de operação;

- em aplicações de negócio (nomeadamente no suporte às actividades), incluindo comunicação interna e externa, processamento e armazenamento de informação, e apresentação e manipulação de informação; aqui se incluem as novas aplicações suportadas pela utilização de comunicações, tipo Internet e Intranet.

## ***6.2 Um conceito potencialmente mobilizador para a indústria e para os serviços***

O mercado potencial dos edifícios inteligentes é, normalmente, associado a grandes edifícios, com mais de 5000 m<sup>2</sup>, utilizados por (pequenas e grandes) empresas envolvidas em trabalhos complexos ou inovadores [DEGW & Teknibank, 95]. O previsto aumento do peso da construção de edifícios de grandes dimensões na Europa até ao fim do século reforça a posição desse segmento de mercado e desta atitude. A razão desta associação liga-se ao facto de, em grandes edifícios, ser possível obter ganhos muito significativos do ponto de vista energético, sendo esta a faceta mais motivadora, do ponto de vista económico, para a adesão ao conceito de edifício inteligente. Corresponde à valorização da primeira componente referida na Tabela 6-2, onde se enfatiza o peso das questões associadas à estrutura do edifício.

Todavia, é opinião do autor que, o mercado dos edifícios inteligentes não se deve restringir aos grandes espaços, sendo do máximo interesse a sua aplicação generalizada, nomeadamente em ambientes residenciais, instituições e empresas, públicas e privadas, no sentido de suportar convenientemente os conceitos emergentes de teletrabalho, redes de empresas e empresas virtuais, bem como melhorar as condições de segurança, quer dos bens, quer das pessoas. Neste enquadramento, é fundamental considerar, de forma equilibrada e articulada, as três componentes referidas na Tabela 6-2.

Por razões naturais (de posicionamento tecnológico) as propostas apresentadas no presente trabalho enquadram-se e enfatizam a importância das segunda e terceira componentes referidas na Tabela 6-2, nomeadamente a aplicação das tecnologias de informação e a disponibilização de tecnologias e serviços integrados.

A abrangência e implicações do desenvolvimento do conceito de edifício inteligente tornam-o como potencialmente mobilizador de um variado leque multidisciplinar de indústrias e serviços. Por indústria, neste caso, entende-se um conjunto de entidades que, para além das empresas industriais tradicionais, incluem micro- e pequenas-empresas nas áreas das tecnologias de informação e consultadoria. Como exemplos de áreas industriais envolvidas refiram-se as de construção, de

comunicações, de sistemas electrónicos, de controlo, de sensores e das tecnologias de informação em sentido lato.

Realça-se o papel que um programa mobilizador nesta área poderia ter, quando se considera a economia nacional, tomando o edifício inteligente não enquanto um projecto de uma empresa ou consórcio de empresas (como é, actualmente, utilizado para ser referido na sede da Caixa Geral de Depósitos, do Centro Cultural de Belém ou da EXPO'98, para utilizar exemplos actuais conhecidos, em particular, pelos habitantes de Lisboa), mas como uma ideia mobilizadora com características adequadas à geração de um elevado e diversificado número de oportunidades de negócio dependentes de projectos específicos, potenciadores da criação de empregos e ganhos de valor acrescentado significativos.

Parece, mesmo, adequado comparar, ainda que de forma parcial e incompleta, as capacidades mobilizadoras dos edifícios inteligentes com as de mega-projectos como a AutoEuropa:

- na produção de veículos automóveis, nomeadamente na Auto-Europa, torna-se necessária uma infra-estrutura centralizadora e integradora das produções de um sem número de empresas industriais dependentes, trazendo problemas logísticos significativos;
- na “produção” de edifícios inteligentes, do ponto de vista das tecnologias de informação, a integração é realizada no cliente final, isto é no edifício, podendo os produtos a integrar ser de tipo físico ou lógico, com problemas logísticos minorados pelos modelos emergentes de redes de empresas e empresas virtuais.

No primeiro caso, o investimento na concepção de um veículo é compensado pelos lucros resultantes da sua comercialização (propriedade industrial); no segundo caso, o investimento realizado na análise dos sistemas e definição dos modelos de referência pode ser de difícil retorno, considerando que a utilização de patentes industriais é, de certa forma, contraproducente em relação à utilização generalizada. De referir, no entanto, a adopção da atitude de “pagar para jogar” adoptada pela Associação Europeia para os Edifícios Inteligentes (tradução de “European Intelligent Building Association”), em relação à divulgação das características do barramento de comunicações proposto por um grupo de operadores industriais (mais à frente, as questões dos barramentos de comunicação serão abordados).

Admitindo, assim, que

**“o conceito de edifício inteligente é mobilizador”,**

importa caracterizar alguns aspectos, como sejam “*quanto* mobilizador pode ser”, “*como* pode ser encarado” ou “*o que* é necessário para que o seja”.

No tocante a *quanto* mobilizador poderá revelar-se, muito depende de *como* seja encarado. Caricaturando a situação, tudo depende se a disponibilidade para envolvimento se pode compara com uma colaboração pontual, um “*part-time*”, um emprego a tempo integral ou para dedicação exclusiva. Uma situação de dedicação exclusiva é, certamente, francamente saudável ou (“simplesmente”) viável para um elevado número de empresas.

No tocante ao *que* é necessário para se constituir em programa mobilizador, consideram-se os seguintes três aspectos:

- utilização extensiva de normas;
- disponibilidade de modelos de referência;
- adopção de uma atitude equilibrada em relação às questões tecnológicas e ergonómicas.

As duas primeiras características estão associadas à perspectiva utilizada nos sistemas abertos, em que se recorre à definição e utilização de modelos, arquitecturas e protocolos de referência. São a garantia de integração sustentada de diversos produtos no ambiente geral dos edifícios inteligentes. As propostas que se apresentam nas secções seguintes pretendem constituir-se nesse conjunto de referência, em termos das arquitecturas a utilizar, no que concerne à área das tecnologias da informação.

A terceira característica enfatiza o peso que o utilizador do edifício deve ter em todas as fases do projecto do edifício, permitindo que, em tempo de exploração do edifício, o utilizador tenha a capacidade de ajustar o modo de interacção com o edifício. Como referido em [Shemie, 97], “um verdadeiro edifício inteligente utiliza a tecnologia para servir, não para dominar”. Na secção seguinte retomar-se-à este assunto.

Em termos meramente indicativos, refere-se, a propósito, o quanto o tema tem sido mobilizador no ambiente académico próximo ao autor, nomeadamente:

- proposta e realização do projecto “ISEnto - Integração de Serviços em Edifícios sensorialmente infra-estruturados”, 1994-96, desenvolvido no UNINOVA e de que resultaram os relatórios técnicos [ISEnto, 94a] a [ISEnto, 96f];
- participação no projecto “SIntEd - Integração de Actividades e Serviços em Edifícios Inteligentes” do programa CYTED, 1996-99, em que o grupo UNINOVA/UNL é proponente e o Prof. A. Steiger Garção responsável pela sua coordenação global;

- participação no projecto PEDIP “ECOCLIMAT - Climatização ecológica”, 1997-2000, na área de desenvolvimento de controladores para equipamento de ar condicionado, em cooperação com a Patrónica e IST, cujos resultados serão integrados nos equipamentos a desenvolver pelo grupo Atecnic;
- no elevado número de projectos de fim de curso que ao longo de vários anos foram seleccionados e realizados por finalistas de Licenciaturas em Engenharia, sob a orientação do autor [Matamouros & Assunção, 93] [Seco & Duarte, 93] [Morgado et al., 94] [Mota & Mendes, 94] [Branco, 95] [Gonçalves, 95] [Vale, 95] [Vivian, 95].

### **6.3 Enquadramento das propostas a apresentar**

No presente trabalho, um edifício inteligente é um ambiente estruturado, equipado com dispositivos sensoriais e sistemas de controlo heterogéneos e vocacionados para a monitoração e controlo de actividades (dentro ou fora do edifício), recorrendo a uma infra-estrutura de redes de comunicação, eventualmente heterogénea, que assegura a comunicação entre utilizadores e sistemas computacionais instalados.

No âmbito dos trabalhos de que esta tese é resultado, foram desenvolvidos alguns modelos de referência que seguidamente serão apresentados.

Nos objectivos iniciais dos trabalhos incluem-se a definição, desenvolvimento e exploração de uma infra-estrutura que permite a integração de serviços e actividades.

Sublinha-se que, as actividades a desenvolver nos edifícios podem ser melhor optimizadas, nomeadamente mais organizadas, mais eficientemente controladas e menos dispendiosas, se um conjunto de infra-estruturas estiver disponível, actuando de modo cooperativo e permitindo a implementação de novas funcionalidades. Estas novas funcionalidades serão obtidas, predominantemente, através da interacção sinérgica dos sistemas disponíveis, no sentido de obter um suporte melhorado às actividades.

Tome-se um exemplo relativo à localização de pessoas no interior de edifícios institucionais ou empresariais. O conhecimento da localização de um membro da organização torna-se, nalgumas situações, da maior importância para a instituição/empresa. A utilização dessa informação pode, por exemplo, permitir o re-encaminhamento automático (isto é, transparente) de chamadas telefónicas<sup>45</sup>,

---

<sup>45</sup> Re-encaminhamento automático de chamadas, tradução de “follow-me”.

originadas no exterior ou no interior da organização, por pessoa qualificada (ou não, dependendo de configuração).

Com este objectivo em vista, diferentes soluções podem ser consideradas. Uma solução dedicada, embora dispendiosa, baseia-se na utilização de transponders<sup>46</sup> ou identificadores pessoais e sistemas com capacidade de localização desses identificadores. Soluções disponíveis no mercado fazem uso de identificadores recorrendo a radiofrequência ou infravermelhos para garantir a comunicação com as unidades de monitoração. Quando o suporte à comunicação é realizado por radiofrequência, o identificador pessoal (eventualmente embebido no cartão de identificação ou no porta-chaves) é detectado quando da passagem do seu portador através de antenas (normalmente colocadas em locais de acesso, nomeadamente embutidos nas ombreiras das portas). Quando o suporte à comunicação é por infravermelhos, os dispositivos pessoais emitem, periodicamente, um código para identificação que é captado por receptores responsáveis pela cobertura de espaços pré-definidos.

Uma solução alternativa, recorrendo à utilização sinérgica de sistemas já disponíveis no edifício, baseia-se na informação disponibilizada pelo sistema de controlo de acessos (quando este permitir uma compartimentação adequada dos espaços sobre controlo).

Com este exemplo, alguns aspectos importantes relativos aos modos de concepção e utilização do sistema podem ser realçados.

O primeiro prende-se com a dicotomia “automático versus solicitado”. Explicitando, em “automático” classificam-se as situações em que a funcionalidade é actividade de forma sistemática, enquanto que em “solicitado” se classificam todas as outras em que o utilizador tem a possibilidade de activar, inibir ou configurar a funcionalidade em causa. Este é um aspecto que se considera da máxima importância. Esta dicotomia necessita de ser resolvida de forma equilibrada em relação aos dois aspectos antagónicos considerados. No caso do exemplo apresentado de re-encaminhamento automático de chamadas telefónicas, independentemente da disponibilidade de tal funcionalidade, deverá ser garantido ao ocupante do edifício utilizá-la ou impedir a sua actuação. De uma forma geral, corresponde a controlar o grau de habilitação de uma funcionalidade em relação a informações/funcionalidades referentes ao utilizador. Obviamente que estas questões ultrapassam a busca de soluções tecnológicas adequadas, sendo necessário recorrer a análises de ordem pessoal e social. Utilizando uma “idéia feita”, comum na “comunidade de edifícios inteligentes”, poder-se-á referir que “as pessoas devem estar primeiro” [Blake-Thomas, 97] [Shemie, 97], em todos os aspectos relativos à concepção, construção, utilização ou exploração dos edifícios. De um modo geral, os

---

<sup>46</sup> “Transponder”, dispositivo com capacidade de comunicação sem fios.

serviços e funcionalidades disponíveis deverão dispôr de vários modos de funcionamento; dependentes do utilizador/sistema solicitante da funcionalidade, bem como do utilizador/sistema alvo, um determinado modo de funcionamento deverá ser seleccionado. Esta característica de equilíbrio entre os aspectos de “automatismo” e de “solicitação” apresenta um forte paralelismo com a área de “automação balanceada”<sup>47</sup> [Camarinha-Matos & Afsarmanesh, 95] [Camarinha-Matos & Afsarmanesh, 96], onde se procura o equilíbrio entre as atitudes antropocêntricas e as estratégias de automação tecnocêntricas em ambientes de produção.

Um segundo aspecto importante resulta da análise de uma outra dicotomia comum relativa a “sistemas distribuídos versus centralizados”. No exemplo apresentado, é fundamental que os sistemas utilizados (central telefónica e sistema de localização) estejam interligados, possuam capacidades de comunicação e consigam comunicar (isto é, que os protocolos utilizados sejam conhecidos e que, directamente ou recorrendo a tradução, o diálogo entre os sistemas seja possível). Para que a funcionalidade “re-encaminhamento automático de chamadas”, como definida, esteja disponível não basta que a central telefónica disponibilize, como é comum em centrais privadas de média e grande capacidade, a facilidade de “follow-me” e que o sistema de localização de pessoas esteja operacional e forneça a informação de posição pretendida em relação a um membro; é fundamental que a informação seja partilhada, de forma aberta (isto é, disponibilizada para quem dela necessitar), entre os dois sistemas envolvidos.

Isto significa que a eficácia associada à adição de novos sensores, dispositivos e sistemas com o objectivo de tornar possível ou otimizar uma acção de controlo ou gestão, é reforçada na medida em que a informação processada possa ser disponibilizada para outros sistemas.

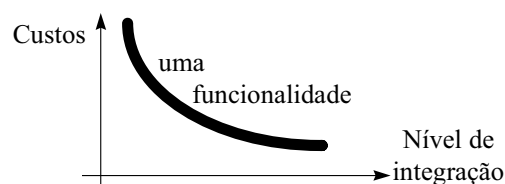


Figura 6-2 - Dependência do custo de uma funcionalidade do nível de integração de sistemas.

É necessário, assim, para além de **integrar** os sistemas presentes no edifício de modo a obter melhorias globais (tais como gestão facilitada e novas funcionalidades e serviços), garantir capacidades da sua adaptabilidade a novos requisitos. Desta forma, o “I” referido no acrónimo EI (ou IB, em inglês), parece aparecer como sendo mais representativo de Integração do que de

<sup>47</sup> “Automação balanceada”, tradução de “balanced automation”.

Inteligente [Gomes & Steiger-Garção, 94]. Em [Arkin & Paciuk, 95], chega a ser discutida de modo introdutório a validade da equação

$$\text{Inteligência do edifício} = \text{Integração de sistemas} \quad (1)$$

De um modo pouco rigoroso, pode afirmar-se que os custos associados à obtenção de uma nova funcionalidade podem ser associados ao nível de integração de sistemas conseguido, como apresentado na Figura 6-2.

Igualmente ilustrativo da importância do nível de integração é o mapeamento de actividades num plano em termos de nível de integração versus o tipo de ocupantes do edifício, considerados em dois grandes grupos, os ocupantes “utilizadores” e os ocupantes “gestores” (isto é, com responsabilidade de gestão sobre algum sistema ou recurso do edifício). Na Figura 6-3, a actividade 1 representa uma actividade específica isolada (isto é, não necessitando de cooperação de sistemas) exclusivamente com interesse para os gestores, enquanto a actividade 2 representa uma actividade predominantemente interessante para os utilizadores, para a qual é necessária “alguma” cooperação entre os sistemas (por exemplo, re-encaminhamento de chamadas telefónicas).

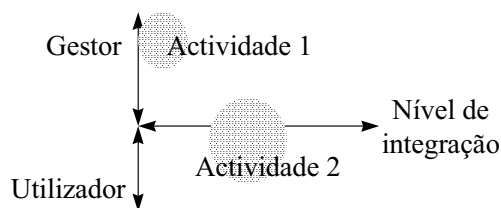


Figura 6-3 - Mapeamento de actividades.

#### **6.4 A necessidade de novos ocupantes: os domots**

A correcta identificação dos diferentes intervenientes nos vários processos presentes nos edifícios é de primordial importância.

Do ponto de vista dos processos, uma classificação comum [DEGW, Arup & Northcroft, 96] considera cinco fases principais:

- concepção,
- projecto,
- construção,
- instalação, e
- operação/gestão.

Do ponto de vista da identificação dos intervenientes, várias classificações são possíveis. Uma caracterização comum, normalmente utilizada por quem está envolvido no projecto e aplicável nas primeiras fases referidas, utiliza uma classificação em quatro grupos de intervenientes:

- arquitectos,
- engenheiros,
- empresários, e
- utilizadores.

Definição alternativa, integrando as últimas fases referidas, recorre a uma caracterização ligeiramente diferente, baseada noutros quatro grupos [Han, 97]:

- arquitectos,
- engenheiros,
- construtores, e
- gestores de infra-estruturas.

No presente trabalho, em que se enfatizam as questões associadas às tecnologias da informação, bem como as últimas duas fases referidas (instalação e operação/gestão), consideram-se três tipos básicos de ocupantes do edifício, nomeadamente:

- utilizadores,
- gestores,
- domots (entidades computacionais a definir seguidamente).

O edifício é, neste trabalho, considerado como um ambiente heterogéneo contendo vários sistemas autónomos e cooperantes e equipado com múltiplos dispositivos sensoriais e de controlo. No sentido de se obter uma eficaz exploração da infra-estrutura instalada é necessário dispôr de uma arquitectura de suporte amigável para o utilizador<sup>48</sup>, permitindo diálogos fáceis e eficientes entre os diferentes ocupantes e a infra-estrutura do edifício.

Desta forma, é necessário dispôr de um conjunto de entidades responsáveis por esses diálogos, com capacidades de gerir recursos e de tornarem essas capacidades disponíveis aos ocupantes. Estas entidades foram, em [Gomes & Steiger-Garção, 96], denominadas por domots<sup>49</sup>, e dispõem das seguintes capacidades:

---

<sup>48</sup> Amigável ao utilizador, tradução de “user-friendly”.

<sup>49</sup> Domot, resultado da concatenação das palavras “domus” e “robot”.

- monitorar o ambiente (no sentido de ter acesso a sensores);
- actuar nos dispositivos da infra-estrutura;
- comunicar com os ocupantes;
- tomar decisões/acções de acordo com as três características anteriores.

Estas capacidades podem ser obtidas autonomamente ou não, isto é, para atingir um objectivo o domot poderá agir autonomamente ou recorrer à cooperação de outro agente. Desta forma, a arquitectura distribuída desenvolvida poderá ser encarada como um sistema multiagente em que os agentes (domots) são vocacionados para o suporte às actividades dos ocupantes do edifício. As características dos domots podem ser enfatizadas em várias componentes, nomeadamente nas suas capacidades de automatizar tarefas, inteligência embebida ou de cooperação integrada, como apresentado na Figura 6-4.

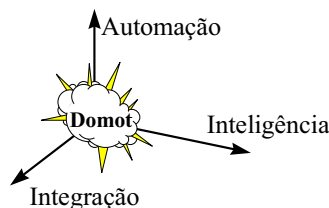


Figura 6-4 - Características principais de um domot.

Em sentido lato, um domot é qualquer entidade (entenda-se mecatrónica ou computacional) actuante no edifício, realizando a ligação entre ocupantes e recursos da infra-estrutura. O seu objectivo primeiro é o de suportar funcionalidades e actividades. Neste sentido, um domot é uma entidade que será encontrada no caminho de um ocupante quando este tenta ter acesso a um recurso ou funcionalidade, ver Figura 6-5.

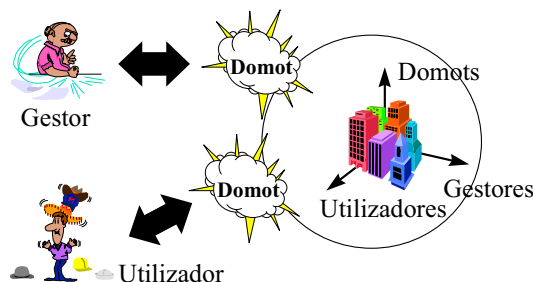


Figura 6-5 - Agentes solicitadores de recursos em edifícios.

Como exemplo de um domot desenvolvido e em operação nas instalações do UNINOVA, refira-se a aplicação Campus-Guard [Gomes et al., 97], a referir mais à frente, vocacionada para a monitoração em tempo-real de um campus de ciência e tecnologia (como o campus da FCT-UNL); dispõe de um

interface de utilizador constituído por um sinóptico apresentando o estado de diferentes dispositivos presentes no edifício, como detectores de presença, câmaras e terminais telefónicos.

Um outro domot, denominado Domot-SIntEd, está sendo desenvolvido no âmbito do projecto CYTED VII.10 SIntEd, permitindo a integração dos sistemas monitorados pelo Campus-Guard com sistemas em desenvolvimento por outros participantes no referido projecto, como sistemas de elevadores, de seguimento visual de pessoas e de reconhecimento de voz.

## 6.5 Perspectivas de modelação

O conceito de edifício inteligente é, necessariamente, multidisciplinar, sendo natural esperar que diferentes comunidades envolvidas nos processos de projecto e exploração tenham diferentes perspectivas sobre os modelos adequados de edifício.

Do ponto de vista deste trabalho, como referido, enfatizam-se as questões de engenharia associadas às tecnologias de informação (excluindo-se as perspectivas associadas a questões de construção, arquitectura e engenharia).

São considerados dois tipos de objectivos para a modelação:

- no primeiro, onde a especificação é o objectivo específico, as características do modelo devem ser traduzidas de um ponto de vista externo (encarando o sistema como uma “caixa preta”);
- no segundo, onde o objectivo específico é a implementação, busca-se uma descrição do sistema em termos das suas características internas.

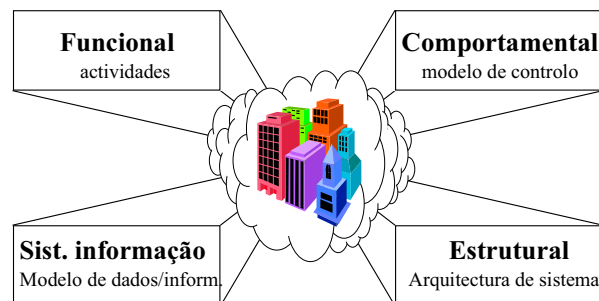


Figura 6-6 - Diferentes perspectivas para a modelação do edifício.

A Figura 6-6 apresenta as quatro principais perspectivas de modelação consideradas: comportamental, funcional, estrutural e sistemas de informação. Devem ser consideradas como vistas complementares,

dando suporte específico às várias actividades de modelação, especificação, implementação e utilização.

A perspectiva “funcional” pode ser vista como um exemplo de “modelar para especificar”, enquanto que as perspectivas “comportamental” e “estrutural” podem ser vistas como exemplos de “modelar para projectar” (entendendo-se projectar em sentido lato como um processo que se estende da especificação à utilização, passando pela implementação).

Na perspectiva “comportamental” estão incluídos os modelos associados ao controlo dos diferentes dispositivos e recursos. Na perspectiva “estrutural” incluem-se os modelos relativos às arquitecturas físicas e lógicas (isto é, *hardware* e *software*). Correspondem às duas perspectivas a que, neste trabalho, se dará mais desenvolvimento. Nas próximas secções serão detalhadas estas duas perspectivas.

A perspectiva “funcional” abrange a descrição das actividades e serviços, em termos de lista de procedimentos ou diagramas de fluxo de dados. Não será detalhada neste trabalho.

Na perspectiva “sistemas de informação” incluem-se os modelos de dados dos sistemas. O modelo de dados utilizado no projecto ISEnto é composto por dois tipos de entidades [ISEnto, 94c] [ISEnto, 95b], genericamente denominadas, respectivamente, por:

- “Instâncias”, e
- “Catálogo”.

A última caracteriza todos os dispositivos e equipamentos disponíveis ou em uso no edifício do ponto de vista das suas características intrínsecas (isto é, de fabrico), enquanto a primeira contém informação específica de todos os dispositivos e sistemas efectivamente instalados.

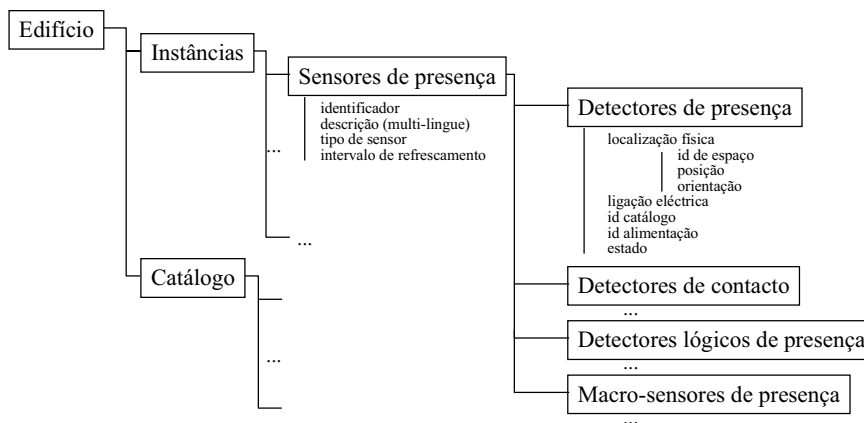


Figura 6-7 - Modelo de dados parcial dos detectores de presença.

Tomando como exemplo um controlador programável, o “Catálogo” é utilizado para reunir as características gerais de fabricante (número e tipo de entradas e saídas, tipo de comunicação, etc.), enquanto que a “Instância” reúne informação sobre como está embebido no sistema (identificador, ligações das entradas e saídas, etc.). Na Figura 6-7 apresenta-se o modelo de dados parcial associado a detectores de presença.

A implementação do modelo de dados utilizado no âmbito do projecto ISEnto baseou-se num modelo relacional, utilizando múltiplas tabelas [ISEnto, 95b], desenvolvidas com o produto da Borland, Paradox Engine (entretanto descontinuado, pelo fabricante). O custo e a disponibilidade de uma biblioteca de funções, acessível a partir de aplicações de utilizador desenvolvidas em C/C++, constituíram-se nos principais critérios para a sua selecção (razão pela qual não se seleccionaram ferramentas baseadas em SQL<sup>50</sup>).

Em [Han, 97] pode ser encontrada a descrição de um trabalho, não muito distante do mencionado, em que as abordagens orientados por objectos são utilizadas.

## **6.6 Ambientes de desenvolvimento e exploração em edifícios**

Considera-se que as actividades de desenvolvimento e exploração associadas à concretização do conceito de edifícios inteligentes se encontram agrupadas em três grupos principais, como apresentado na Figura 6-8. Estes três grupos de actividades estão associados a grupos de pessoas, infra-estruturas computacionais e tempos de reacção completamente distintos e são genericamente referidos como ambientes de desenvolvimento, de configuração e de execução, respectivamente [Gomes et al., 94a] [Gomes & Steiger-Garção, 96a] [Gomes et al., 97].

O ambiente de desenvolvimento suporta, em tempo de projecto, a integração dos resultados dos diferentes ambientes computacionais e grupos de projecto.

O ambiente de execução é responsável pela exploração da infra-estrutura disponível no edifício.

O ambiente de configuração, fazendo a “ponte” entre os dois, permite que os resultados do ambiente de desenvolvimento sejam instalados na planta do edifício (isto é, nos sistemas efectivamente instalados no edifício) de forma segura e transparente para os ocupantes.

Pretende-se, deste modo, suportar as atitudes de implementação incremental de soluções, imprescindíveis a um suporte efectivo à mudança e re-utilização de infra-estruturas. A capacidade de

---

<sup>50</sup> SQL - Structured Query Language

adaptação à mudança, da responsabilidade dos gestores de infra-estruturas<sup>51</sup>, fica, desta forma, implicitamente embebida na metodologia de desenvolvimento utilizada.

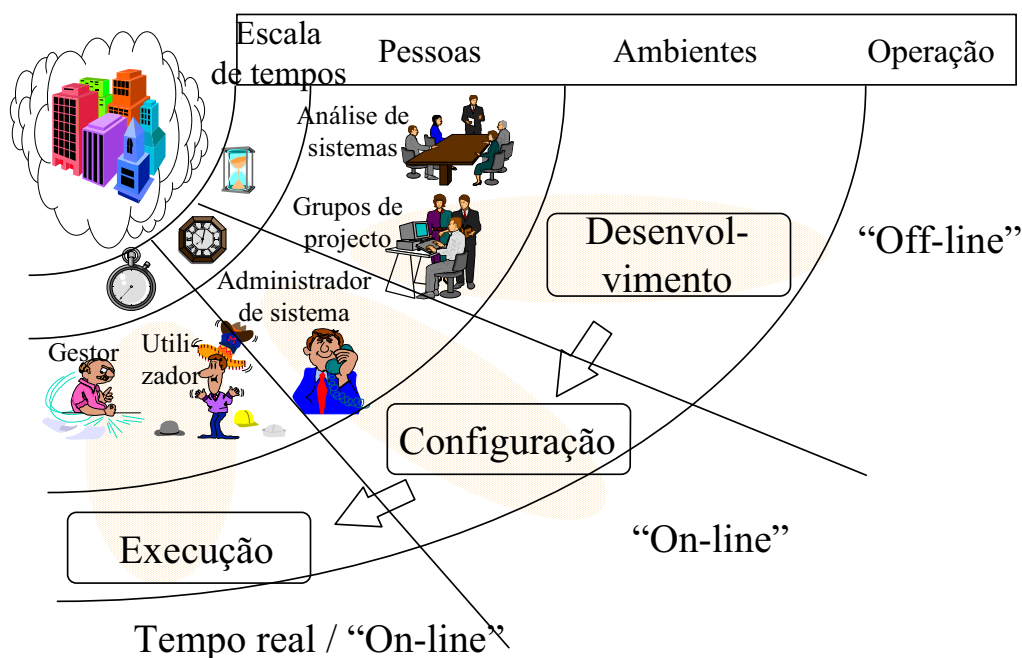


Figura 6-8 - Atividades de desenvolvimento decorrentes do conceito de edifício inteligente.

As atribuições do ambiente de configuração, vitais para o suporte efectivo às actividades e funcionalidades disponíveis, nomeadamente às capacidades de adaptação à mudança, são:

- garantir um registo e seguimento das versões instaladas do sistema;
- assegurar uma fácil actualização de versões e transição entre versões;
- assegurar uma instalação fácil de novos sistemas;
- assegurar a reconfiguração de recursos (decorrente da gestão da infra-estrutura).

O objectivo do ambiente de desenvolvimento, referido no topo da Figura 6-8, é o de actuar como integrador dos resultados produzidos por diferentes grupos de projecto, como se evidencia na Figura 6-9.

Estes grupos de projecto intervêm nas diferentes áreas, desde o projecto de estruturas e construção (arquitectos e engenheiros civis), até às infra-estruturas eléctricas e electrónicas (energia, AVAC<sup>52</sup>, iluminação, comunicações de voz, imagem e dados), até ao projecto dos diferentes sistemas a instalar.

<sup>51</sup> “Gestão de infraestruturas”, tradução de “facility management”.

<sup>52</sup> AVAC, acrónimo para “Aquecimento, Ventilação e Ar Condicionado”.

Do ponto de vista do grupo de projectistas associados à “parte inteligente”, importa garantir que as informações provenientes das diferentes ferramentas de apoio a projecto, genericamente designadas por ferramentas CAD<sup>53</sup>, sejam integráveis, de forma a contribuírem para a representação global do edifício. Deste modo, os resultados produzidos por CADs geométricos (projecto de arquitectura do edifício), CADs eléctricos (projectos de energia, AVAC, iluminação, telefones, redes, sistemas de automação, etc.) e CADs específicos (projectos de controlo, incluindo configuração de sistemas) são utilizados como entradas para o ambiente de desenvolvimento, responsável por produzir informação que permita instalar (através do ambiente de configuração) os ambientes de execução, isto é, exploração da infra-estrutura.

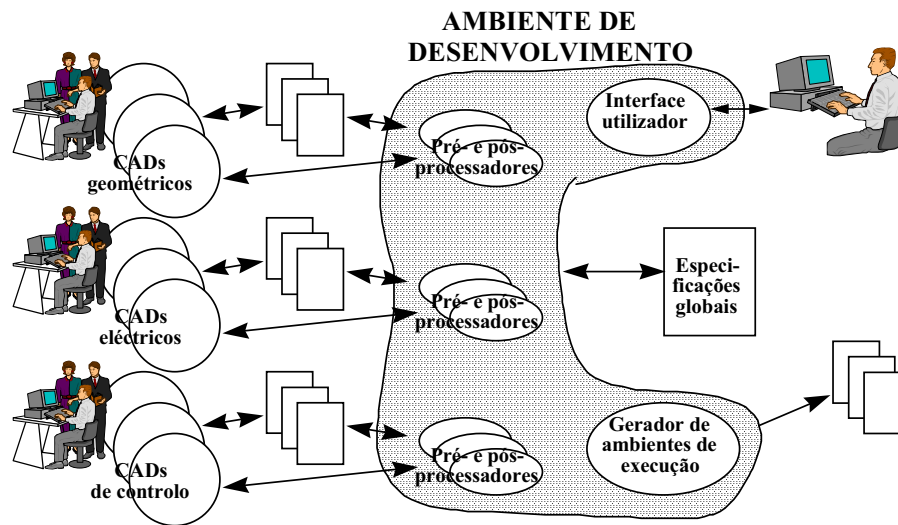


Figura 6-9 - Caracterização do ambiente de desenvolvimento.

É nesta fase de integração que se gera o modelo de dados global do edifício. Como exemplo, retorne-se à Figura 6-7, para referir que a informação constituinte do conceito de “detector de presença” é colectada de vários CADs, nomeadamente “localização” e “orientação” do CAD geométrico, “interligação eléctrica” do CAD eléctrico, etc..

Dois protótipos foram realizados como partes constituintes deste ambiente de desenvolvimento:

- o primeiro, resultado de um projecto de fim de Licenciatura em Engenharia Informática [Morgado et al., 94], permite utilizar uma aplicação largamente utilizada, o AutoCAD da Autodesk, como ambiente de especificação de todas as características geométricas necessárias, quer da estrutura do edifício, quer da localização dos dispositivos e sistemas. Como resultado, essas características geométricas são armazenadas numa base de dados, à

<sup>53</sup> CAD, acrónimo de “Computer Aided Design”, isto é, Projecto (ou Desenho) Assistido por Computador

qual os módulos responsáveis pela execução têm acesso. O já referido produto Paradox-Engine da Borland foi utilizado no seu desenvolvimento, permitindo a utilização de bases de dados geradas pelo produto Paradox da Borland, bem como por outras aplicações que também recorram ao Paradox-Engine (ou compatível);

- o segundo, resultado de trabalho de estágio no âmbito do projecto ISEnto, permite editar directamente a base de dados referida, obviando a ausência de vários módulos de interligação a outros CADs.

## **6.7 Arquitecturas de referência**

Como referido, considera-se fundamental para a capacidade mobilizadora do conceito de edifício inteligente, a existência de modelos, arquitecturas e protocolos de referência aplicáveis ao ambiente de execução, isto é, associados à operação efectiva no edifício. Importa completar esse conjunto de referência com as arquitecturas propostas quer para a interligação física de dispositivos e sistemas (arquitectura *hardware*), quer para a interligação lógica das aplicações e processos computacionais (arquitectura *software*).

Ambas as arquitecturas baseiam-se em modelos de referência largamente utilizados, quer no meio académico, quer no meio empresarial. Nas subsecções seguintes procede-se à sua apresentação.

### **6.7.1 Interligação de sistemas e dispositivos**

#### *6.7.1.a Redes e protocolos comuns*

Sendo o conceito de sistema aberto um dos paradigmas utilizados como referência, inicia-se a presente secção com a identificação de algumas arquitecturas de interligação e protocolos comumente utilizados para garantir a integração de sistemas em ambientes prediais.

Um número significativo de protocolos de comunicação actualmente em uso são protocolos proprietários; outros são protocolos abertos e outros, ainda, são disponibilizados aos membros das instituições promotoras. É consensual que nenhum dos protocolos disponíveis tem condições para vir a ser utilizado como um protocolo universal, pelo menos no futuro próximo, sendo claros alguns dos posicionamentos das empresas e instituições com intervenção na área em torno de alguns dos protocolos. A título de exemplo, refiram-se:

- o BACnet como reunindo algum consenso na indústria americana e indústria “não alinhada”,
- o EIB tendo papel semelhante em relação à indústria europeia e
- as redes IP em relação a instituições académicas e de investigação.

Qualquer das soluções se enquadra no paradigma dos sistemas abertos; no entanto, enquanto que o BACnet e o protocolo IP estão em domínio público, dispondo de normas internacionais associadas, o EIB utiliza uma filosofia de “pagar para jogar”, típico do modo como foi proposto (por um grupo de empresas, que o utilizam nos seus produtos comerciais).

De entre os protocolos mais comuns refiram-se os seguintes [HPAC, 96] [Yomura, 97]:

- o BACnet (Building Automation and Control Network), desenvolvido pela ASHRAE (American Society of Heating, Refrigeration and Air-Conditioning Engineers), foi aprovado como norma ANSI/ASHRAE 135-1995 e está em processo de aprovação no âmbito da ISO (International Standards Organization). É um protocolo de comunicação de dados, vocacionado para a gestão integrada dos sistemas de automação predial. Tem como objectivo o suporte à “interoperação” dos diferentes sistemas e dispositivos, isto é, à integração de sistemas de automação predial. Para isso, os diferentes componentes devem ter capacidade de trocar mensagens de acordo com uma linguagem definida na referida norma BACnet, em que se utilizam extensivamente os conceitos de “objecto” (definem-se 18 tipos normalizados), “propriedades” (identificam-se 123 tipos) e “serviços” (definem-se 32 tipos de serviços, agrupados em cinco grupos). Ao nível do suporte físico à comunicação, cinco tipos de redes de dados são possíveis:
  - Ethernet (ISO/IEC 8802-3), com taxas de transferência de dados da ordem de 10 a 100Mbps e com um custo elevado;
  - ARCNET (ATA/ANSI 878.1), com taxas de transferência de dados da ordem de 0,156 a 10Mbps e com um custo médio/elevado;
  - MS/TP, isto é, “Master Slave / Token Passing” (ANSI/ASHRAE 135-1995), utilizando a norma EIA-485, permitindo ligações multiponto, com taxas de transferência de dados da ordem de 9.6 a 78.4 Kbps e com um custo baixo;
  - Ponto-a-ponto (EIA-232), permitindo a comunicação entre dois dispositivos específicos;

- LonTalk, desenvolvido pela Echelon e que mais à frente se caracterizará, com taxas de transferência de dados da ordem de 4,8 a 1250 Kbps e com um custo variável;
- o CEBus (Consumer Electronics Bus), desenvolvido pela EIA (Electronic Industries Association) e formalizado na sua norma EIA-600, teve como objectivo inicial a automação de residências. Pode utilizar diversos suportes físicos à comunicação: rede de distribuição de energia, cabo coaxial, par entrançado, infravermelhos e radiofrequência;
- o EIB (European Installation Bus), originalmente desenvolvido por um grupo de companhias liderado pela Siemens, é mantido, actualmente, pela European Installation Bus Association. A rede de interligação pode, para além de garantir a comunicação de dados (2 fios), fornecer a alimentação aos dispositivos (3 fios, isto é, alimentação e terra de protecção);
- o LonTalk foi desenvolvido pela Echelon, empresa dos E.U.A., e é destinado a um espectro largo de aplicações prediais. Está implementado num circuito integrado específico, denominado “neuron chip” e é o suporte para a arquitectura genericamente designada por LonWorks, da Echelon. A comunicação pode ser suportada por diversos meios físicos, nomeadamente par entrançado, cabo coaxial, rede de distribuição de energia, radiofrequência, infravermelhos e fibra óptica;
- as redes de campo (designação geral para os normalmente referidos “*field-buses*”) destinadas a ser utilizadas em diferentes áreas de aplicação, onde se inclui, para além da automação predial, a automação industrial (possivelmente a sua maior área de aplicação). O ProfiBus e o WorldFip são dois exemplos significativos (referidos na norma europeia EN50170);
- o X-10, largamente utilizado nas actividades de controlo de residências nos E.U.A., com uma ênfase muito forte na facilidade de instalação pelo “curioso”, permite a utilização de meios de comunicação baratos, como redes multiponto e a rede de distribuição de energia instalada na residência;
- o N2, desenvolvido pela Johnson Controls, Inc., E.U.A., é baseado num protocolo muito simples (ASCII) e numa rede RS-485 a três fios, com um máximo de 100 dispositivos. Utiliza uma disciplina de mestre-escravo.

Excluíram-se de uma referência individualizada protocolos com “alguma” utilização, como o BatiBus - Building Bus (francês), o CAB - Canadian Automated Building Protocol (canadiano), o Data

Highway Plus (utilizado pela Allen-Bradley), o FND - Firm-Neutral Data Transmission (alemão) e o Modbus (suportado por elevado número de fabricantes de controladores programáveis industriais), entre outros.

De entre os protocolos referidos, alguns foram propostos para utilização restrita em ambientes residenciais, como o X-10 e o CEBus, enquanto outros estão vocacionados para a integração de sistemas em edifícios institucionais, como o BACnet e o EIB.

### 6.7.1.b Arquitectura desenvolvida

Em [Gomes et al., 95] [Gomes & Steiger-Garção, 96a] [Gomes et al., 97] foi proposta e explorada uma arquitectura distribuída, baseada em dois níveis de suporte à interligação de sistemas em edifícios, como apresentado na Figura 6-10.

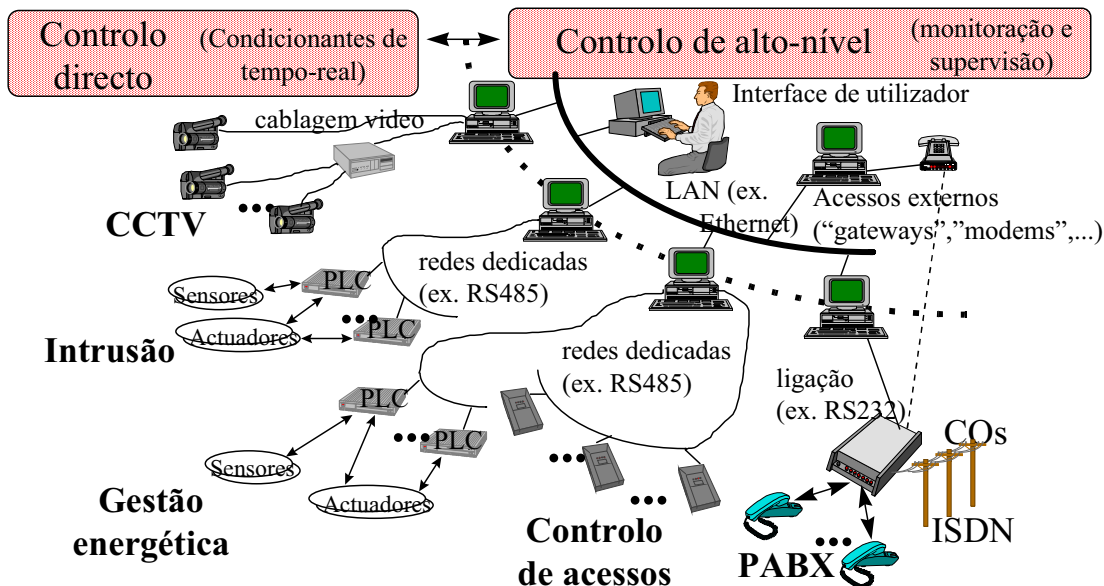


Figura 6-10 - Arquitectura lógica de referência para interligação física de dispositivos e sistemas.

O nível de topo é vocacionado para a interligação de estações de monitoração e supervisão, responsáveis pelo controlo de alto-nível da infra-estrutura instalada. Este nível é constituído por uma rede de alto débito utilizando o protocolo TCP-IP<sup>54</sup> e com capacidades de interligação a outras redes, quer através de Ethernet, quer através de outras ligações satisfazendo o protocolo TCP-IP (por exemplo PPP sobre um canal “telefónico” série). De entre as vantagens mais significativas da utilização do protocolo TCP-IP refiram-se as seguintes:

<sup>54</sup> TCP-IP, acrónimo de “Transfer Control Protocol - Internet Protocol”.

- protocolo normalizado, com suporte na totalidade das plataformas de interesse;
- suporte implícito para modelo cliente-servidor;
- gestão transparente para o utilizador da localização das estações e dos meios físicos de suporte à comunicação.

O nível inferior é vocacionado para a interligação de sistemas específicos, nomeadamente controladores, responsáveis pelo controlo local da infra-estrutura instalada. Este nível é, tipicamente, constituído por redes locais dedicadas de baixo débito, nomeadamente RS-232 e RS-485. A gestão de cada uma das redes constituídas é da responsabilidade de uma estação de monitoração e supervisão, que, por sua vez, dispõe de capacidades para disponibilizar a informação para toda a estrutura.

É uma arquitectura de referência simples e utilizada largamente em ambientes de computação distribuídos, nomeadamente naqueles que são vocacionados para a monitoração e controlo de processos industriais. De entre estes, especial referência para os ambientes de produção, onde o “know-how” existente pode ser re-utilizado e aplicado quase que directamente no contexto dos edifícios inteligentes. Como caso particular, mencionam-se as redes de campo (“field-buses”) brevemente referidas na subsecção precedente, vocacionadas para a interligação de controladores, dispositivos sensoriais e de actuação, que podem ser utilizadas, com vantagens, como suporte alternativo para a rede de nível inferior mencionada.

Algumas características pretendidas, nomeadamente de flexibilidade para suportar instalações incrementais, são intrinsecamente suportadas por esta arquitectura; por exemplo, adicionar uma estação numa rede TCP-IP é uma tarefa (normalmente) tranquila.

Realce-se a estrutura intrinsecamente hierárquica de acesso à informação, em que para se conhecer o estado de um determinado sensor é necessário ganhar acesso à estação de monitoração que será responsável pelo diálogo com o controlador específico associado ao sensor em causa.

Os sistemas referidos na Figura 6-10, são-no a título exemplificativo, ilustrando as áreas em que mais atenção tem sido dada em termos da produção de protótipos. Concretamente, os componentes principais do protótipo de domot de monitoração, vocacionado para a segurança das instalações, denominado Campus-Guard [Gomes et al., 97], integra os sistemas de intrusão, de vídeo e de interligação à central telefónica,

As áreas de gestão energética (nomeadamente nos sistemas de iluminação e AVAC) e de controlo de acessos perfilam-se como as áreas que irão receber no futuro próximo desenvolvimento de domots específicos. No caso do controlo de acessos, de referir o interesse de integração de técnicas avançadas

de identificação que permitam automatizar os processos de validação de identidade, actualmente realizados com base em cartões magnéticos ou transponders. A introdução de técnicas baseadas em características pessoais, como a impressão digital, o rosto, a iris ou a voz, entre outras, afigura-se como de elevado interesse. Alguns trabalhos em curso na UNL-FCT e no UNINOVA colocam-se nesta linha estratégica, nomeadamente os trabalhos no âmbito do já referido projecto SIntEd, em que o Campus-Guard será utilizado como ponto de partida para a integração dos referidos sistemas, aos quais se adicionarão sistemas de controlo e monitoração de elevadores.

Para deixar clara uma ideia forte do modelo de referência em análise, que fica pouco evidente na Figura 6-10, refira-se que o controlo de um qualquer sistema específico não necessita de estar centralizado numa estação de controlo de alto-nível; poderá ser distribuído na rede da forma mais adequada para a aplicação específica em causa.

Tome-se o exemplo dos sistemas de intrusão e de monitoração video, em operação num campus adequadamente infra-estruturado, do ponto de vista de comunicações (isto é, com redes locais de dados). As soluções tradicionais adoptam uma concepção fortemente centralizadora, passando pelo cabeamento específico para todas as câmaras do campus a partir de uma central de segurança, considerada como ponto de concentração de todos esses cabos. A solução aqui apresentada baseia-se na aquisição, processamento, compactação e transmissão da imagem digitalizada através de uma rede de alto débito disponível. É, inclusivamente, aconselhável a utilização da rede Ethernet da instituição para esse fim, dado que a largura de banda dessa rede está normalmente disponível durante o período em que é necessário realizar a monitoração das instalações, isto é, fora das horas de serviço. Deste modo, as cablagens para sinais video ficam compartimentadas em áreas “pequenas” (relativamente à situação inicial), com o associado impacto económico, e a infra-estrutura de redes do campus pode ser utilizada de forma mais eficiente. As necessidades de interligação resultantes de situações de crescimento ou divisão do campus e correspondente reestruturação da rede IP, são, do ponto de vista do sistema de segurança, resolvidos “transparentemente” (considera-se que a ligação IP é tão natural como a ligação à rede de águas, por exemplo).

Esta solução comporta, no entanto, alguns compromissos que não devem ser descurados, nomeadamente:

- sendo a transmissão de video e voz, como tarefas com fortes restrições de tempo-real, realizada, tradicionalmente, com recurso à utilização de uma largura de banda fixa (por exemplo, em comunicações telefónicas ou de video-conferência), a solução utilizada baseada num protocolo intrinsecamente sem capacidades de garantir qualquer largura de banda para uma aplicação específica, pode, em situações de sobrecarga, ver o seu desempenho degradado significativamente. Esta situação pode ser obviada através da

caracterização do pior caso verificável, em termos do tráfego possível e posterior dimensionamento conveniente das redes;

- sendo possível que o tráfego gerado por qualquer das estações limite a utilização do meio de comunicação pelas outras estações ligadas no mesmo troço, existe a possibilidade de “empastelamento” das comunicações por uma estação “hostil”, o que, do ponto de vista de segurança, é, sem dúvida, uma situação de vulnerabilidade eventualmente intolerável.

Qualquer das situações descritas pode ser solucionada através da utilização de troços específicos para garantir os débitos e isolamento necessários.

### **6.7.2 Interligação lógica**

Em [Gomes & Steiger-Garção, 94] foi proposta uma arquitectura distribuída como referência para a interligação entre processos das estações de controlo (entendendo-se controlo em sentido lato como monitoração e supervisão), posteriormente actualizada em [Gomes et al., 95] [Gomes & Steiger-Garção, 96a] e [Gomes et al., 97].

A arquitectura é baseada numa estrutura cliente-servidor e suportada por comunicação interprocessos através de “sockets-TCP”, como apresentado na Figura 6-11.

Esta arquitectura permite a interligação de estações heterogéneas, incluindo estações de baixo custo e de elevado desempenho, sendo apenas necessário satisfazer o referido protocolo de intercomunicação. A ligação aos processos de controlo local, associados a equipamento específico, é garantida através da utilização de conversores de protocolo, por exemplo, de rede-TCP-IP para rede-RS-485, e de “gateways” específicos.

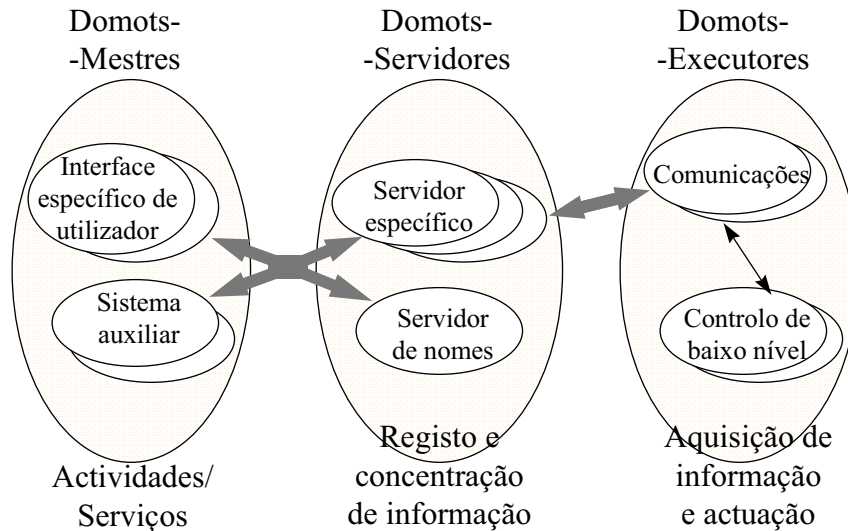


Figura 6-11 - Arquitectura lógica de referência para interligação de processos em estações de controlo.

O conjunto de processos em execução é dividido em três grupos, a saber:

- grupo de processos de aquisição de informação e actuação, denominados genericamente por Domots-Executores, associado aos processos de controlo local de baixo-nível, responsáveis pela actuação de dispositivos e aquisição de informação e em execução em controladores dedicados; o acesso a estes processos é realizado através de servidores de comunicações, responsáveis pela conversão de protocolos (de que o conversor rede-RS-485 para rede-TCP-IP dispõe de um protótipo desenvolvido e em operação, e que conversores de rede-TCP-IP para outros protocolos específicos, como o BACnet, podem vir a receber desenvolvimentos no futuro próximo);
- grupo de processos de registo e concentração de informação, denominados genericamente por Domots-Servidores, constituindo o núcleo do sistema e composto por um conjunto de servidores associados à monitoração de recursos específicos; são responsáveis por fornecerem informação actualizada sobre o estado dos recursos monitorados (num mecanismo cliente-servidor) e de garantirem o armazenamento (a longo-termo) da informação; podem ser classificados em dois grupos:
  - servidores específicos, responsáveis pela gestão de um determinado tipo de recurso; como exemplos de servidores com protótipos instalados referem-se os servidores de sensores, de imagens e de central telefónica; exemplos de servidores a desenvolver incluem os servidores de controlo de acessos e de gestão energética; num sistema poderão existir qualquer número de servidores específicos;

- servidores auxiliares, nomeadamente o servidor de nomes, com função semelhante a um servidor de nomes num domínio TCP e responsável por fornecer informação (nomeadamente nome, estação e porto de serviço) acerca de todas as entidades existentes no sistema, desde sensores, dispositivos, equipamentos, espaços, etc.; a replicação do servidor de nomes deve ser considerada com cuidado, sendo necessário garantir a coerência da informação nas várias instâncias;
- grupo de aplicações de negócio, actividades e serviços, denominados genericamente por Domots-Mestres associado às actividades de controlo de alto-nível, monitoração e supervisão, e de modo impreciso à interacção dos domots com os ocupantes humanos do edifício; incluídos neste grupo encontram-se:
  - interfaces de utilizador, vocacionados para permitir a gestão de recursos específicos ou fornecer suporte para a análise de actividades; como exemplo, refira-se, novamente, o domot Campus-Guard que se encontra instalado e é vocacionado para a vigilância de instalações;
  - sistemas auxiliares, nomeadamente sistemas periciais e de controlo automatizado de processos, responsáveis pela emissão de alarmes e recomendações ou pelo controlo autónomo de recursos.

Assim, os processos associados a cada um dos três níveis referidos são classificados como diferentes variedades de domots, de acordo com a Tabela 6-3.

|                                      |                   |
|--------------------------------------|-------------------|
| Aplicações de negócio                | Domots-Mestres    |
| Registo e concentração de informação | Domots-Servidores |
| Aquisição de informação e actuação   | Domots-Executores |

Tabela 6-3 - Classificação dos processos em execução.

A Figura 6-12 apresenta, a título exemplificativo, uma distribuição de processos, testada nas instalações do UNINOVA [Gomes et al., 97], utilizando três estações (uma, PCU, executando o interface de utilizador e o servidor de nomes e as restantes duas, PCS-1 e PCS-2, executando servidores específicos), uma central telefónica, PABX (interligada por linha série RS-232 a um servidor/conversor de protocolos de comunicações) e uma rede de três controladores programáveis, PLC1 a 3 (responsáveis pelo controlo local dos detectores de presença do sistema de intrusão).

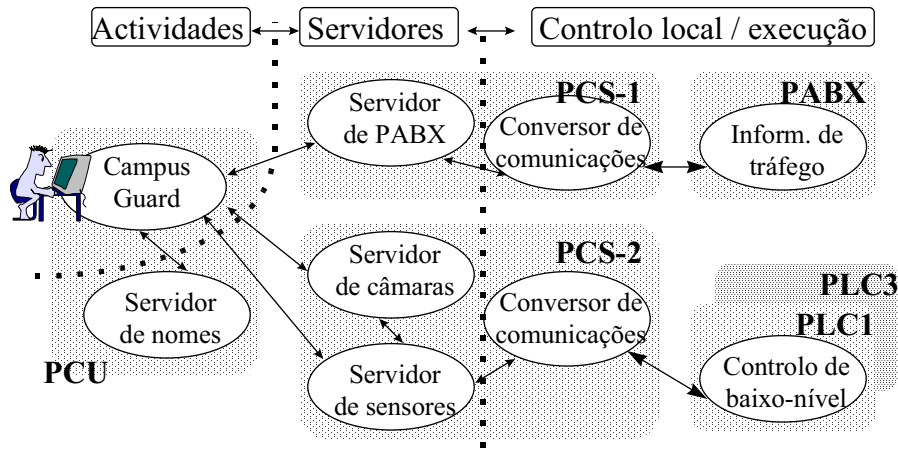


Figura 6-12 - Configuração laboratorial de teste.

### 6.7.3 Caracterização dos serviços disponíveis

A estruturação proposta do ambiente de exploração da infra-estrutura instalada no edifício, nomeadamente através da sua estrutura básica cliente-servidor, revela-se como facilmente generalizável e aplicável a outros ambientes onde a supervisão, monitoração e controlo de processos, quer locais quer remotos, seja pretendida, nomeadamente a sistemas SCADA. De referir que a disponibilidade de protótipos parciais do sistema foi um factor importante e motivador do convite para participação no projecto europeu Esprit IV 21017 - INNOVA “High Performance Telecontrol Station with Image and Sound Capabilities”, iniciado em Abril de 1996, onde está prevista a adaptação e integração de um interface de utilizador adequado a um sistema de telecontrolo avançado, com utilização integrada de sistemas sensoriais complexos, como visão e som, objecto de desenvolvimento do referido projecto.

A portabilidade deste tipo de soluções passa pelo conhecimento, numa base de sistema aberto, dos protocolos e formatos de mensagem utilizados no estabelecimento das comunicações interprocessos. Seguidamente são funcional e resumidamente descritos dois dos servidores específicos desenvolvidos, concretamente o servidor de imagens e de sensores. Descrições mais detalhadas podem ser encontradas na documentação produzida no âmbito do projecto ISEnto “Integração de Serviços em Edifícios sensorialmente infra-estruturados” [ISEnto, 94a] a [ISEnto, 94d] [ISEnto, 95a] a [ISEnto, 95c] [ISEnto, 96a] a [ISEnto, 96f].

O diálogo com os servidores é, de forma geral, garantido através de mensagens de texto, isto é, os pedidos são textuais enquanto as respostas podem ter informação em binário embebida (caso de imagens, por exemplo, em que as questões de compactação de informação são determinantes). Desta forma, a utilização de um servidor, bem como de algumas das tarefas de depuração associadas, pode ser realizada através de um terminal comum com a comunicação suportada por *sockets* TCP.

O conjunto de serviços disponibilizados pelos servidores específicos pode ser caracterizado através da divisão em serviços comuns a todos os servidores e serviços específicos de cada servidor.

A apresentação destes conjuntos de serviços é realizada seguidamente. Exceptuam-se desta apresentação:

- o servidor de central telefónica [ISEnto, 96e], que fornece um conjunto de serviços simples relativos a lista telefónica e de apoio à produção de relatórios de tráfego telefónico por terminal telefónico e centro de custos existentes;
- o servidor de comunicações [ISEnto, 96b], estabelecendo a ponte entre uma rede TCP/IP e a rede dedicada de controladores, sendo responsável pela gestão do diálogo com os controladores, nomeadamente recuperação de erros por retransmissão, “*time-outs*”, etc;
- o servidor de nomes [ISEnto, 96a], responsável por fornecer informação sobre todas as entidades definidas no sistema, nomeadamente a localização do servidor apto a fornecer informação específica sobre a entidade.

### 6.7.3.a *Serviços comuns*

O conjunto de serviços disponíveis comuns a todos os servidores divide-se em três grupos de serviços:

1. o primeiro serviço (“*Características\_de\_Servidor*”) fornece a assinatura do servidor em causa;
2. outros dois serviços (“*Altera\_Língua*” e “*Informa\_Sobre\_Língua\_Em\_Uso*”) destinam-se a suportar uma utilização multilingue do sistema. Através deles é possível obter as descrições de todas as entidades utilizadas numa língua à escolha, que, na versão actual pode ser Português ou Inglês. De notar que esta é uma característica associada a cada ligação cliente-servidor estabelecida, podendo existir num mesmo servidor pedidos de clientes em línguas diferentes, isto é, são suportados concomitantemente interfaces de utilizador utilizando línguas diferentes;
3. outra funcionalidade comum é fornecida pelo serviço de palavras-chave (*passwords*), encarregue da segurança básica dos servidores, isto é, só permite acesso aos serviços disponíveis a clientes que, aquando do estabelecimento da ligação, enviem uma palavra-chave válida. O serviço disponibilizado (“*Altera\_Código\_de\_Acesso*”) permite alteração da palavra-chave.

### 6.7.3.b *Serviços específicos do servidor de imagens*

O servidor de imagens permite controlar as câmaras e fornecer, numa base cliente-servidor, informações associadas a imagens, nomeadamente a sua representação digital tendo em conta resolução e dinâmica de intensidades pré-definidas. O conjunto de serviços disponíveis específicos ao servidor de imagens divide-se em três grupos de serviços:

1. os três primeiros serviços (“Lê\_Imagem\_Uma\_Vez”, “Monitorar\_Imagem”, “Esquecer\_Imagem”) referem-se a actividades de aquisição de imagens. O serviço “Monitorar\_Imagem” permite activar um ciclo de refrescamento sobre uma imagem, com um período característico e com envio periódico de informação sobre a imagem ou da própria imagem, que pode ser efectuado, caso o cliente o pretenda, apenas quando a imagem é “suficientemente diferente” da última imagem enviada. A quantificação da diferença é realizada através do número de pixels “diferentes” em duas imagens consecutivas (isto é, adquiridas em instantes separados pelo período característico), sendo considerado um valor (parametrizável) de diferença (“*threshold*”) entre valores de dois píxeis para que estes se considerem diferentes;
2. através do segundo grupo de serviços (“Controlar\_Câmara” e “Estado\_de\_Câmara”), é possível o posicionamento, actuação nos circuitos de auto-íris, focagem e zoom das câmaras, selecção ou definição de posições de posicionamento, quando disponíveis. Estas operações de controlo são implementadas através de pedidos ao servidor de sensores;
3. finalmente, através do serviço “Características\_de\_Imagem”, um cliente poderá obter informação diversa sobre uma imagem, nomeadamente informação sobre as suas dimensões.

Do ponto de vista deste servidor, o conceito associado a “imagem” corresponde a uma matriz rectangular de pixels, em que um pixel pode representar um nível de cinzento ou uma cor através das suas componentes RGB. Uma “imagem” pode estar associada a uma imagem física ou a uma imagem lógica, resultante de pós-processamento de uma outra imagem.

Na categoria das imagens físicas agrupam-se as que são capturadas directamente pelos sistemas de aquisição de imagem, isto é, resultando directamente de um processo de aquisição operada no sinal video fornecido pela câmara. Na categoria das imagens lógicas ou pós-processadas, estão aquelas que são obtidas como resultado de processamento sobre imagens já existentes (físicas ou pós-processadas). Estes processamentos permitem, no presente, obter imagens que são uma região de outra imagem ou obter imagens de menor resolução.

### 6.7.3.c *Serviços específicos ao servidor de sensores*

O servidor de sensores, também responsável pelos actuadores presentes no sistema, permite conhecer os estados dos diversos tipos de sensores físicos e lógicos, bem como actuar em saídas binárias do sistema de controladores. Estas saídas são utilizadas para actuação de câmaras e alimentação de sensores e dispositivos; no futuro, serão também utilizadas para controlo de acessos, de iluminação, de ar-condicionado, etc. Tanto a leitura do estado dos sensores físicos como a actuação das saídas é feita recorrendo ao servidor de comunicações que gere o diálogo de baixo nível com os controladores, lendo os valores presentes nas entradas dos PLCs e actuando nas suas saídas.

É suportado o conceito de sensor lógico, isto é, de virtualização de sensores, resultado do processamento da informação sensorial de um dispositivo.

Como exemplo elementar de sensor lógico, actualmente implementado no servidor de imagens por questões de eficiência e baseado no serviço “Monitorar\_Imagem”, refira-se o sensor binário de presença com base em diferenças de imagens.

O conjunto de serviços disponíveis específicos ao servidor de sensores divide-se em três grupos de serviços:

1. os três primeiros serviços (“Ler\_Estado\_de\_Sensor“, “Monitorar\_Sensor“, “Esquecer\_Sensor“) referem-se a actividades de leitura do estado dos sensores. Enquanto o serviço “Ler\_Estado\_de\_Sensor” fornece como resposta o estado actual do sensor especificado, o serviço “Monitorar\_Sensor” permite manter uma actividade de leitura desse estado em intervalos regulares. A exemplo do servidor de imagens, no sentido de minorar a comunicação supérflua, o cliente pode optar por ser notificado apenas quando houver alteração do estado do sensor, garantindo o servidor, no entanto, uma actividade de refrescamento da informação sensorial com o período especificado nas tabelas de configuração;
2. os três serviços seguintes (“Actuar\_Saída“, “Desactuar\_Saída“, “Reset\_de\_Dispositivo“) dizem respeito a actividades de actuação nas saídas. A actuação pode ser imediata (ligar ou desligar) ou temporizada, dependendo da configuração do actuador, armazenada em base de dados [ISEnto, 94c] [ISEnto, 95b]. Está ainda disponível um serviço de reinicialização (“reset“) de dispositivo, conduzindo a uma desactivação temporária da saída;
3. finalmente, através do serviço “Características\_de\_Sensor“, um cliente poderá obter informação diversa sobre os sensores existentes, nomeadamente quais os sensores

existentes (identificadores e descrições associadas), por tipos (de presença, de contacto ou lógicos) e por localização (edifício, sala, coordenadas).

#### 6.7.4 Exploração: o exemplo do Campus-Guard

Como aplicação-piloto das arquitecturas de referência apresentadas na Figura 6-10 e na Figura 6-11 foi desenvolvido um primeiro domot, denominado Campus-Guard [Gomes et al., 97]. Consiste num sinóptico vocacionado para a monitoração de espaços em que os diferentes dispositivos sensoriais presentes no edifício são apresentados e é possível obter informação específica em relação a qualquer desses dispositivos representados. Foi utilizado um sinóptico organizado hierarquicamente, proposto em [Gomes & Steiger-Garção, 94] e que se apresenta na Figura 6-13.

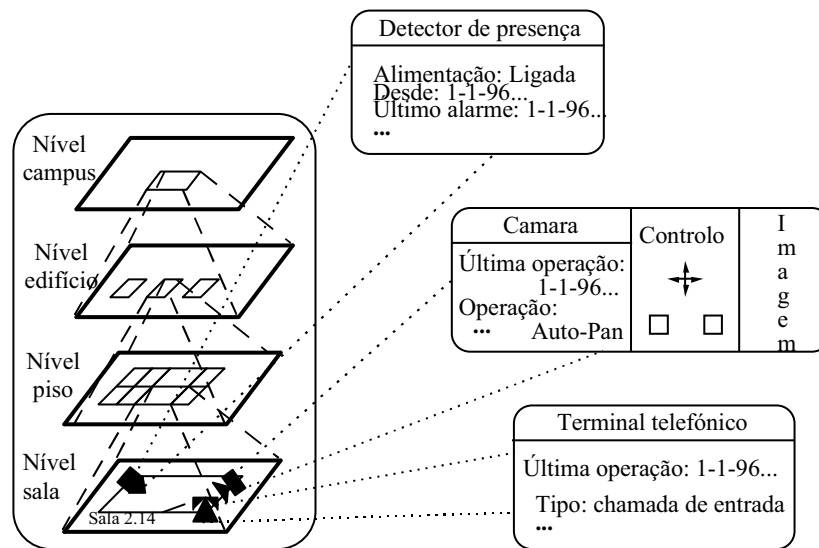


Figura 6-13 - Estruturação do sinóptico de monitoração de instalações.

Na Figura 6-14 apresenta-se um interface típico relativo à utilização do Campus-Guard na monitoração das instalações onde o trabalho foi desenvolvido, no Centro de Robótica Inteligente do UNINOVA. A caracterização geométrica dos espaços foi previamente obtida utilizando o editor AutoCAD da AutoDesk e um resultado de [Morgado et al., 94].

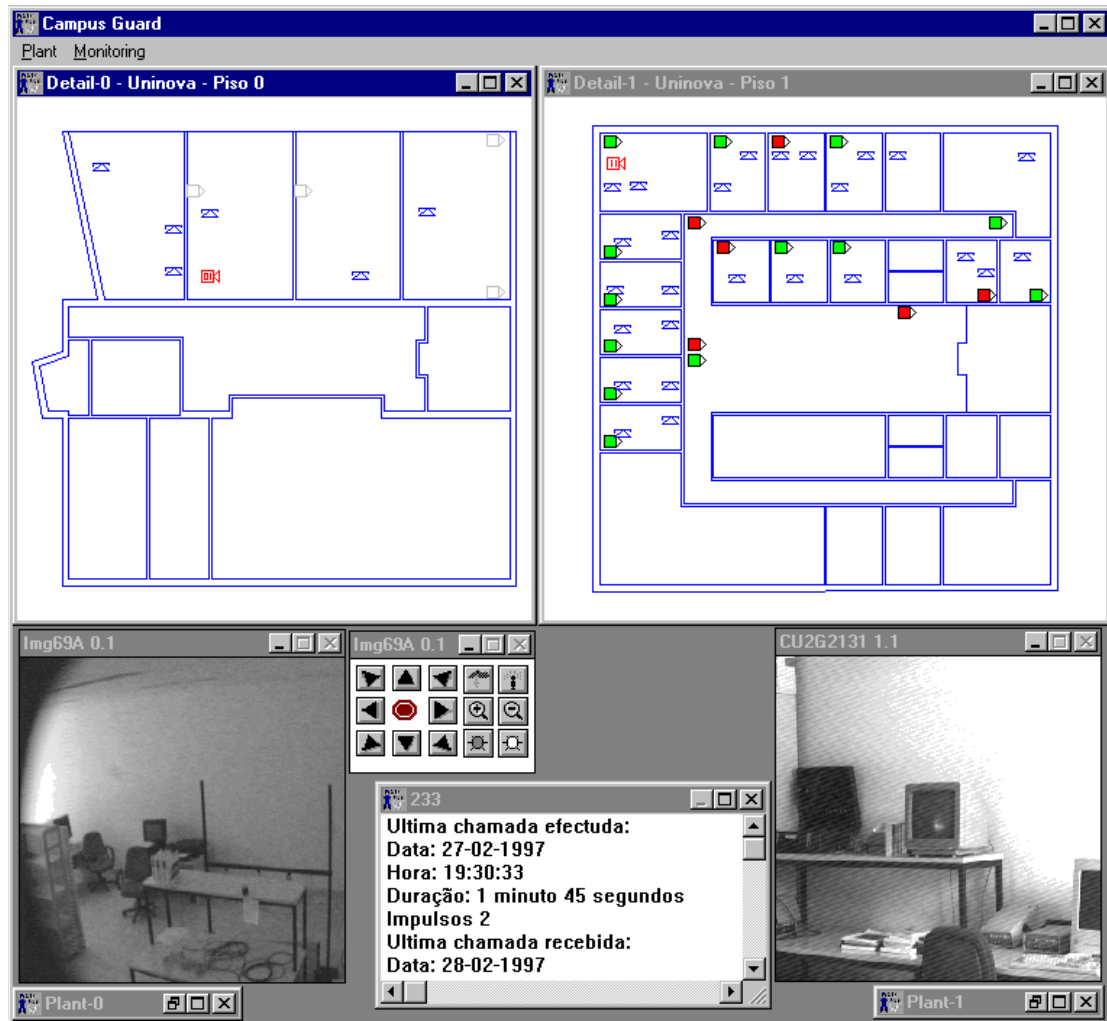


Figura 6-14 - Interface típico do Campus-Guard.

A estrutura do Campus-Guard permite uma adaptação em tempo de execução ao ambiente que se pretende monitorar, dado que a informação apresentada é obtida dinamicamente, isto é, em tempo de execução. O servidor de nomes é consultado para obter informação sobre os dispositivos instalados nos vários espaços, bem como sobre a localização dos servidores específicos responsáveis pela sua monitoração. Posteriormente, cada servidor específico é contactado, fornecendo a informação a ser visualizada.

É possível obter informação detalhada sobre os vários dispositivos representados no sinóptico (numa atitude de modelação orientada por objectos), nomeadamente:

- a representação dos detectores de presença utiliza um código de cores representando o seu estado actual (desligado, ligado, activo, desconhecido, ...);
- as imagens referentes às várias câmaras podem ser visualizadas, bem como os comandos de posicionamento mecânico associados;

- em relação a cada extensão telefónica é possível obter informação específica sobre o tráfego associado.

## **6.8 Análise de casos**

Como referido, o conceito de edifício inteligente e a área de domótica constituíram-se em fonte de aplicações e de justificação de requisitos para as classes das RdP-R e RdP-RH.

Os edifícios inteligentes são um ambiente complexo, de operação intrinsecamente distribuída e fortemente heterogéneo e que não dispõem de ambientes abertos e integrados de suporte ao projecto, com capacidades de integrar especificações e resultados de diferentes grupos de projecto actuantes nesta área de aplicações.

No que se relaciona com a área de contribuição das tecnologias de informação, no particular da especificação e implementação dos subsistemas, bem como na sua validação e verificação, as RdP-R e as RdP-RH apresentam-se como formalismos com características muito adequadas para serem utilizadas pelos ambientes de desenvolvimento, aplicáveis nas várias fases do seu ciclo de vida.

A sua utilização como formalismo intermédio reforça a afirmação anterior, permitindo a tradução de outros formalismos e consequente integração das especificações produzidas, de acordo com a natureza do problema e dos recursos e conhecimento disponíveis para a sua resolução. Quer os statecharts, quer formalismos de controlo inteligente, formalismos de amplo espectro de aplicação e interesse, são traduzíveis na nova classe de RdP, como demonstrado nos capítulos anteriores.

Complementando os exemplos apresentados e a identificação de algumas lacunas nos suportes teóricos de análise e implementação das RdP-R, nas subsecções seguintes identificar-se-ão algumas pontes potenciais entre as RdP-R e RdP-RH e os trabalhos desenvolvidos para a monitoração, controlo e gestão integrados (apresentados nas secções precedentes), que se perspectivam como interessantes para receber investigação adicional.

### **6.8.1 Modelação de estruturas sensoriais e de visualização**

As questões de integração e fusão de informação proveniente de sensores tem sido alvo de extensos e diversificados trabalhos. Como trabalho relevante no levantamento do estado da arte, refira-se [Luo & Kay, 89]. Em várias arquitecturas de processamento de informação sensorial, como os sensores lógicos, utilizam-se extensivamente os conceitos de hierarquia e de estruturação modular.

Em linha com esses trabalhos, as RdP-R apresentam-se como um formalismo adequado para a modelação de estruturas sensoriais.

Tome-se um exemplo associado ao sistema de segurança, especificamente a detecção de uma situação de alarme. Na especificação dos alarmes é comum fazer uso de dependências entre os vários sensores presentes, traduzidas através de expressões lógicas e sequências pré-definidas, como na programação de controladores lógicos programáveis (isto é, o sistema é caracterizável por uma máquina de estados ou por um circuito combinatório). Em tempo de projecto é realizada a especificação dos diferentes alarmes de interesse, através de estruturas específicas. As RdP-R podem ser utilizadas para representar estas estruturas.

Enfatiza-se o facto deste modelo poder ser utilizado não apenas para, em tempo de especificação e validação, suportar a análise do modelo através de simulação ou de outros métodos formais, mas também para ser directamente utilizado, em tempo de execução.

Considere-se uma sala, referida por *lab1*, com quatro sensores binários, nomeadamente um detector de fecho de fechadura (*sensorA*), um detector de presença volumétrico (*sensorB*), um detector de movimento baseado em ultra-sons (*sensorC*) e uma barreira de infravermelhos (*sensorD*). Considere-se o caso da condição de alarme ser expressa através de uma expressão lógica, enunciada, por exemplo, através da seguinte regra:

IF (sensorA OR (sensorB AND sensorC) OR sensorD) THEN alarmONlab1

A tradução para um modelo RdP-R de uma regra como a indicada é directa, como apresentado em capítulo anterior. O estado de cada um dos detectores utilizados pode ser representado por uma marca presente num lugar específico, sendo as dependências expressas na regra modeladas através da RdP-R da Figura 6-15.

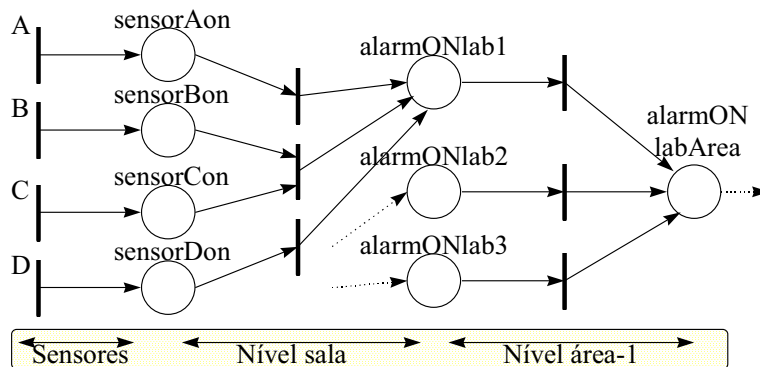


Figura 6-15 - Modelo RdP-R simplificado de uma estrutura de alarme.

Na Figura 6-15 inclui-se a representação de uma segunda regra, hierarquicamente superior à apresentada, modelando a detecção de um alarme numa área que inclui a referida sala. A estrutura de alarme é modelada através da seguinte regra:

```
IF (alarmONlab1 OR alarmONlab2 OR alarmONlab3) THEN alarmONlabArea
```

A relação entre a marcação dos vários lugares utilizados e das entidades representadas no sinóptico hierárquico da Figura 6-13 pode ser directamente realizada, de modo que o sinóptico se comporte como uma representação da RdP-R, cuja execução tem reflexo directo na animação desse sinóptico, isto é, no seu estado (por exemplo, na Figura 6-14).

Do ponto de vista de análise formal do modelo apresentado na Figura 6-15, em que os sensores são representados através de transições de entrada, a rede é não limitada; deste modo, torna-se necessário completar o modelo com as condicionantes necessárias para que a rede seja limitada. Isso é garantido através da inclusão de um lugar que “fecha” a rede, a exemplo do que foi realizado no capítulo anterior em relação ao modelo de controlador difuso. Como descrito no capítulo anterior, esse lugar é marcado no início do ciclo de análise e será utilizado como lugar de entrada para as transições representativas das entradas sensoriais externas, complementado pela sincronização das transições com o evento “início de ciclo de análise”.

Uma necessidade complementar do ponto de vista da modelação do sistema é a de facilidade de gestão da complexidade. No caso vertente, a substituição dos detectores de presença e de movimento (*sensorB* e *sensorC*, respectivamente) por um sistema de processamento de imagens, como o servidor de imagens atrás referido, com capacidades de, com base na comparação de duas imagens consecutivas, fornecer uma informação lógica sobre a detecção de intrusos, constitui um exemplo. A modelação deste tipo de sensor lógico pode ser realizada simplesmente através da substituição da dependência nas variáveis *sensorB* e *sensorC* por uma dependência de uma variável *sensorE*, obtida através de função que tenha acesso ao servidor referido; no entanto, do ponto de vista do modelo, o nível de simplicidade/complexidade é mantido.

Complementarmente e como referido em capítulo anterior, uma forma alternativa de suporte à gestão da complexidade, utiliza as capacidade de justaposição (“*folding*”) naturais às RdP Coloridas, de forma geral e às RdP-R, em particular.

Continuando a utilizar o exemplo da estrutura de alarme, considere-se a substituição do detector binário de fecho de fechadura, referido por *sensorA*, por um sistema de controlo de acessos com capacidade de identificação de utilizadores, referido por *sensorF* (através da leitura de cartões, *transponders* pessoais ou identificação de impressões digitais, por exemplo). Neste caso a saída do

sistema sensorial passa a ser representável por uma estrutura que inclui um código de identificação do utilizador. A sua referência no modelo RdP-R é possível através da substituição da dependência em *sensorA* para uma dependência em *sensorF(pessoa)*, em que o parâmetro “*pessoa*” pode ser utilizado nas características da rede autónoma, nomeadamente nos atributos das cores das marcas utilizadas, bem como nas guardas associadas às transições, permitindo implementar “filtros” adequados a emissão de alarmes unicamente nas situações em que o utilizador “*pessoa*” não esteja autorizado a ter acesso ao espaço em causa (*lab1*, no caso do exemplo).

Complementarmente, a potencial relação entre modelos RdP-R e a modelação de interfaces gráficos de utilizador ficou claramente exposta nos parágrafos anteriores. No entanto, essa relação pode ser muito mais estreita e ser estendida à própria modelação do interface, considerando os seus vários componentes, dinâmica e condicionantes mútuos. Constitui-se, inclusivamente, como um forma de verificar as propriedades do interface especificado. Esse é o objectivo de trabalhos significativos desenvolvidos em Toulouse utilizando RdP de Alto-Nível; veja-se, por exemplo, [Sibertin-Blanc, Hameurlain & Touzeau 95].

No particular das RdP-R, considerando a ligação possível entre as características autónomas e não-autónomas do modelo, a sua aplicação à modelação deste tipo de sistemas apresenta-se como de elevado interesse (e pronta a receber trabalho de investigação adicional). Concretamente, prevê-se a sua utilização na especificação e desenvolvimento do software do terminal de operação da estação de controlo de processos resultante do projecto ESPRIT INNOVA (trabalhos actualmente em curso).

### 6.8.2 Modelação de sistemas distribuídos

Outra área de aplicação natural das RdP com particular relevância em edifícios é a de modelação de sistemas distribuídos. As referências a trabalhos sobre a modelação através de RdP de sistemas distribuídos são frequentes na literatura. Refiram-se, a propósito, as publicações de Wolfgang Reisig, como excelentes textos didácticos (considere-se, por exemplo, [Reisig, 92] como texto introdutório geral).

No particular das arquitecturas de referência apresentadas, uma lacuna significativa foi detectada ao nível da modelação e implementação dos servidores: capacidade de verificação formal do seu correcto funcionamento. Embora no desenvolvimento dos protótipos referidos, tenham sido utilizadas as regras comuns de boa programação e de teste, nada garante que novos erros não venham a ser detectados.

A utilização de RdP-R na sua especificação, permitindo proceder à sua verificação e à geração automática de código directamente a partir da especificação produzida, pode contribuir, de modo significativo, para a obtenção de software “verificado”, mais robusto e documentado. Condições

adequadas que permitam desenvolvimentos específicos adicionais nessa direcção tentarão ser reunidas no futuro próximo.

Todavia, de referir que, de um modo geral, as propostas apresentadas nesta tese beneficiaram as implementações centralizadas, em detrimento da execução distribuída das especificações RdP-R.

É, certamente, um tema com impacto primordial em domótica em que (também) se considera necessário e plenamente justificado trabalho de investigação adicional.

Nessa linha de trabalhos, de referir o ambiente de desenvolvimento de aplicações para arquitecturas concorrentes denominado CAB (Concurrent Applications Builder) [Bruno et al., 95], suportado pela linguagem Protob [Baldassari & Bruno, 88].

Nesse ambiente de desenvolvimento recorre-se a uma modelação baseada em objectos e a interacção entre os objectos é realizada através do envio e recepção de marcas. Os lugares envolvidos na interacção com o exterior constituem o interface do objecto. Existem lugares (ou portos) de entrada e de saída. Os lugares de entrada receberão marcas resultado das comunicações de outros objectos, isto é, são marcados de forma autónoma em relação ao modelo explicitamente representado; a notação gráfica utilizada representa-os através de um círculo com linha dupla. Quando um objecto pretende comunicar com outro objecto, envia uma marca para o outro objecto, colocando-a num lugar de saída. Um lugar de saída é representado através de um círculo com um triângulo inscrito. Um lugar de saída não conserva marcas, dado que se considera que são imediatamente enviadas para o(s) objecto(s) de destino.

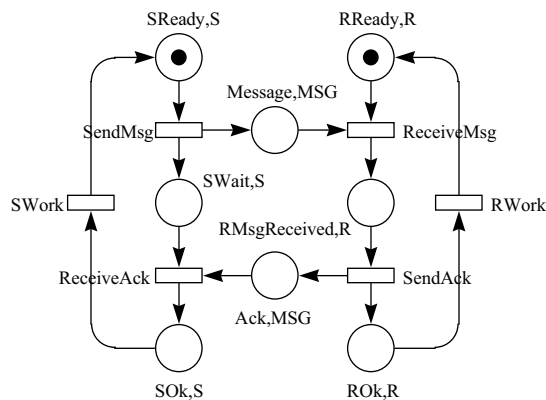


Figura 6-16 - Interação entre um emissor e um receptor.

As Figura 6-16 e Figura 6-17 apresentam um exemplo académico típico, modelando a interacção entre um emissor e um receptor (ou um servidor e um cliente, se se pretender), extraído de [Bruno et al., 95] onde o protocolo TCP-IP é responsável pelo suporte à comunicação entre processos servidores específicos.

Na Figura 6-16, as duas entidades do sistema são modeladas conjuntamente, enquanto que na Figura 6-17 se apresentam os modelos adequados a uma implementação distribuída do sistema.

Na Figura 6-17(a) os módulos constituintes (representados como quadrados com linha dupla) e respectivas ligações são esquematizados, enquanto nas Figura 6-17(b) e Figura 6-17(c) se representam os modelos de cada um dos objectos referidos. Neles se faz uso dos lugares-porto referidos de forma suficientemente clara, poupando explicações adicionais.

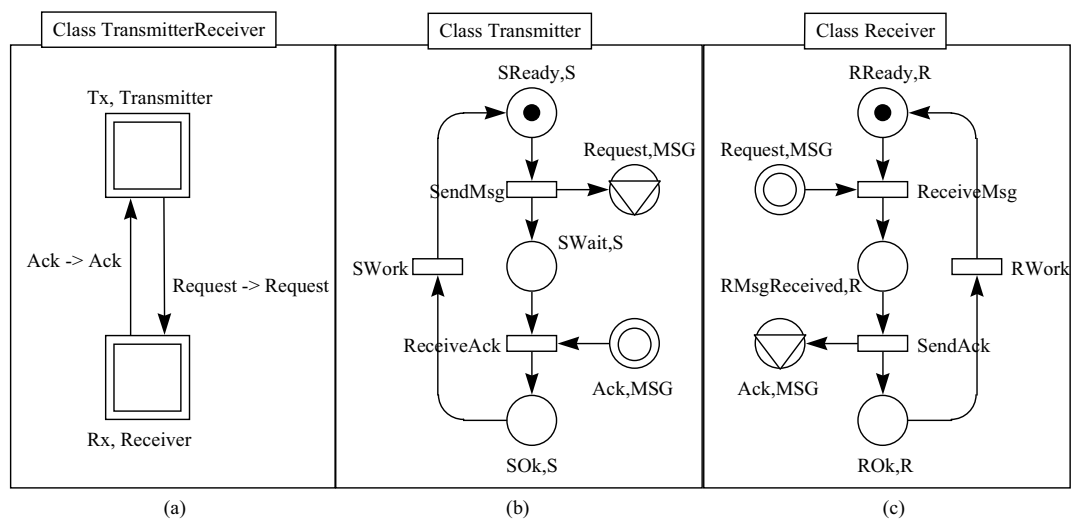


Figura 6-17 - Decomposição em objectos: (a) interação, (b) emissor e (c) receptor.

De referir a elevada proximidade de objectivos entre o ambiente CAB agora descrito e os das RdP-R; desta forma, a comparação de resultados e de soluções para aplicações semelhantes apresenta-se como de elevado interesse para ser conduzida no futuro.



## 7. Conclusões

*Quem feio ama, bonito lhe parece*  
(*Provérbio popular*)

### *Sumário do capítulo*

No presente capítulo pretende-se fazer um balanço do trabalho realizado e apresentado nesta tese, em termos da verificação do nível de cumprimento dos objectivos inicialmente identificados, de uma apreciação crítica sobre as potencialidades e limitações dos modelos propostos e das direcções que se apresentam interessantes para continuar o trabalho ou para o complementar.

Deste modo, o capítulo é dividido em três secções:

- 7.1 Sobre o trabalho realizado*
- 7.2 Sobre os resultados obtidos*
- 7.3 Sobre o trabalho futuro*



## 7. Conclusões

### 7.1 Sobre o trabalho realizado

O balanço do trabalho realizado tendo esta tese como motor ou enquadramento é, necessariamente, multifacetado. Identificam-se, naturalmente, várias linhas de actividades em que será conveniente alguma reflexão sobre o trabalho realizado. Nos parágrafos seguintes abordam-se três aspectos:

- i) nível de divulgação e utilização das temáticas abordadas na tese pela comunidade científica em Portugal, isto é, o enquadramento nacional dos trabalhos;
- ii) publicação de resultados;
- iii) participação em projectos e de transferência de resultados.

#### 7.1.1 Enquadramento nacional

No que se refere à apresentação de trabalhos com RdP em conferências e eventos similares realizados em Portugal, o cenário que se verificava alguns anos atrás era muito pobre: uma ou duas comunicações mais ou menos “desgarradas” era o máximo que se observava, por norma. Progressivamente, o cenário tem melhorado: em 1996, no CONTROLO’96, organizado (também) pela Universidade do Porto, em 1997, quer no ISIE’97, organizado (também) pela Universidade do Minho, quer no AARTC’97, organizado (também) pela Universidade do Algarve, várias comunicações sobre RdP foram apresentadas. Em 1998, por iniciativa do autor desta tese, realizar-se-á a ICATPN’98 (“A” Conferência da Comunidade de Redes de Petri) na região de Lisboa, com a organização da Universidade Nova de Lisboa e do UNINOVA. É convicção do autor que estas são indicações suficientemente claras e seguras que permitem antever que o panorama no tocante à utilização de Redes de Petri em Portugal na próxima década será significativamente diferente do verificado na década anterior.

Situações distintas são as verificadas nas áreas dos edifícios inteligentes, em que a motivação tem sido marcadamente “de oportunidades de negócio” ou fora das áreas das tecnologias de informação, e na

do controlo difuso em que existe um volume de trabalho científico significativo e regular há algum tempo (e que se prevê que seja mantido).

### **7.1.2 Publicação de resultados**

No tocante ao segundo aspecto referido e em completa sintonia com a prática corrente no Grupo de Investigação em que o trabalho se desenvolveu, promoveu-se a apresentação de resultados preliminares em conferências ou workshops da especialidade, à medida que iam estando disponíveis. É-me grato verificar que todas as submissões de trabalhos relacionados com esta tese foram aceites e que da sua apresentação pública resultaram alguns desenvolvimentos muito gratificantes.

A evolução no número de publicações registou a maior incremento no ano de 1995; é, aparentemente, uma evolução normal num trabalho que se realizou por um período ligeiramente superior ao que é recomendável (pela Lei Portuguesa), isto é, acima de cinco anos (se considerarmos que a primeira comunicação foi preparada e publicada ainda em 1992).

A incidência dos conteúdos das várias publicações é realizada de forma aproximadamente uniforme em relação aos vários capítulos, com excepção do capítulo 4, por coincidência o mais longo e o que tem menos trabalho publicado.

### **7.1.3 Participação em projectos e transferência de resultados**

No que se refere ao terceiro aspecto, duas componentes devem ser consideradas: interligação a projectos e transferência de resultados.

Em relação à primeira componente, participação em projectos, a situação que se observava no início dos trabalhos sofreu uma alteração drástica. De facto, no início dos trabalhos desta tese, o autor não estava envolvido em nenhum projecto com perfil adequado para receber os seus resultados ou para suportar a sua evolução (dado não existirem no Grupo de Investigação em que os trabalhos se desenvolveram). A situação actual é, felizmente, completamente distinta, estando em curso três projectos em que as temáticas abordadas na tese são assuntos fundamentais para os trabalhos previstos:

- no primeiro, um projecto de cooperação pré-competitiva no âmbito do programa CYTED, denominado “SIntEd - Integração de Sistemas e Actividades em Edifícios Inteligentes”, que decorre entre 1996 e 1999, em que para além da temática dos edifícios inteligentes constituir o centro integrador de actividades, as RdP e o controlo difuso encontram um forum de discussão entre os vários Grupos de Investigação participantes;

- no segundo, um projecto suportado pela Medida 4.4 B “Projectos Mobilizadores para o Desenvolvimento Tecnológico” do programa PEDIP e denominado “ECOCLIMAT - Climatização ecológica”, que decorre entre 1997 e 2000, em que a temática do controlo difuso e dos sistemas a eventos discretos será abordada no contexto do controlo de equipamentos inovadores de climatização e ar condicionado;
- finalmente, um terceiro, no âmbito do programa ESPRIT, denominado “INNOVA - High Performance Telecontrol Station with Image and Sound Capabilities”, que decorre entre 1996 e 1998, em que as questões do projecto e dimensionamento de sistemas de tempo-real, por um lado, e a modelação de interfaces com o utilizador, por outro, são áreas de potencial utilização das RdP; lamentavelmente não será possível, em relação ao primeiro aspecto, envolver o consórcio e aplicar os resultados obtidos.

Os três projectos encontram-se, desta forma, em áreas de penetração complementares, com ênfases distintas, combinando parceiros industriais (com perfis de produção e de investigação aplicada e desenvolvimento) com parceiros académicos e de institutos de transferência de tecnologia.

De referir ainda os trabalhos realizados no âmbito do projecto ISEnto, já terminados e desenvolvidos no UNINOVA, que permitiram a preparação e maturação das participações nos referidos três projectos.

Em relação à segunda componente mencionada, a transferência de resultados, refiram-se as contribuições para o curriculum da Licenciatura em Engenharia Electrotécnica da FCT-UNL, em particular da disciplina de Síntese e Análise de Sistemas Digitais, bem como para trabalhos integrados em teses de mestrado, projectos de fim de curso e estágios profissionais (já referidos ao longo da tese, quando adequado).

## **7.2 Sobre os resultados obtidos**

Os objectivos enunciados no capítulo inicial para o presente trabalho foram, globalmente, atingidos, pese, embora, o facto de não ter sido implementado (completamente) um ambiente integrado de suporte ao desenvolvimento com RdP-R e RdP-RH. A atitude de princípio seguida no trabalho de investigação de procurar as sinergias resultantes dos avanços verificados em três aspectos distintos do trabalho (teoria, ferramentas e aplicações) revelou-se frutífera.

As classes de RdP propostas foram utilizadas com sucesso ao longo desta tese como suporte à integração de especificações produzidas com diversos formalismos, nomeadamente statecharts, sistemas de regras e controlo difuso. Esse era, recorda-se, um dos principais objectivos do trabalho.

Como resultados daí directamente decorrentes, refiram-se:

- aumentar os graus de liberdade que o projectista de sistemas reactivos de tempo-real dispõe, na medida em que lhe é permitido seleccionar um formalismo, considerado como mais adequado para modelar subsistemas específicos;
- a possibilidade de aplicar aos modelos produzidos nesses formalismos, as capacidades de verificação de propriedades aplicáveis às RdP-R e RdP-RH. Esta é uma questão muito sensível para o caso particular dos statecharts, que não dispõem de ambientes abertos para suporte a trabalhos de engenharia; desta forma, as RdP-R e RdP-RH permitem, potencialmente, proceder à verificação de especificações produzidas em statecharts.

Um outro resultado importante do trabalho está relacionado com as RdP Coloridas e trabalhos associados, em particular com as RdP Coloridas Hierárquicas e RdP Coloridas Temporizadas. Como referido, as RdP Coloridas são uma das classes de referência das RdP Reactivas. Em particular, os modelos de RdP Coloridas podem ser consideradas como casos particulares de modelos de RdP Reactivas.

Importa referir que todos os resultados obtidos para as RdP Coloridas são aplicáveis para as RdP Reactivas, na mesma medida que os resultados das RdP Lugar-Transição são aplicáveis às RdP Interpretadas (ou seja, na medida em que se condiciona a execução das RdP com características dos sistemas físicos reais).

Comparando as generalizações realizadas em relação às RdP Coloridas para obter as RdP Reactivas e Hierárquicas, vários são os resultados mais satisfatórios que os obtidos em relação às realizadas para obter as RdP Coloridas Hierárquicas e RdP Coloridas Temporizadas, nomeadamente:

- as soluções propostas para a estruturação hierárquica dos modelos são mais gerais e “equilibradas”. Por um lado são disponibilizadas capacidades de modelação hierárquica ao nível da execução, com a utilização de metanós, que, dado recorrerem às características não-autónomas do modelo, são de todo inviáveis para modelos autónomos, como o das RdP Coloridas Hierárquicas. Por outro lado, ao nível da modelação envolvendo unicamente características autónomas, é possível utilizar macronós de vários tipos (macrotransições, macrolugares e macroblocos), permitindo obter modelos mais compactos e “legíveis”, ao invés das RdP Coloridas Hierárquicas em que unicamente as macrotransições são permitidas;

- as soluções propostas para a modelação de condicionantes temporais permitem utilizar o espaço de estados em funções de verificação e de execução do modelo (enquanto nas RdP Coloridas Temporizadas apenas a verificação é contemplada).

As vantagens daí resultantes são óbvias; no trabalho esperado para o futuro próximo, tentar-se-ão potenciar essas capacidades.

### **7.3 Sobre o trabalho futuro**

Embora estando antecipadamente precavido para a prática, corrente e necessária, em trabalhos de investigação de ir tomando as opções à medida que o trabalho se vai desenrolando, no espírito do alerta do poeta quando diz que “o caminho se faz caminhando”, nos parágrafos seguintes resumem-se alguns temas em que se considera justificado trabalho adicional de investigação e/ou de desenvolvimento.

Consideram-se duas vertentes:

- a primeira associada a tarefas de “consolidação” do trabalho apresentado;
- a segunda associada a trabalhos que podem ser suportados no trabalho apresentado.

Nas tarefas de “consolidação” incluem-se as que permitam obter um ambiente de desenvolvimento aberto e integrado, suportando as várias fases de desenvolvimento dos sistemas, de acordo com a Figura 3-3 da página 74. É, fundamentalmente, um trabalho de grupo, que se baseia na alteração e integração de alguns trabalhos e protótipos actualmente disponíveis, como os resultantes de trabalhos de Mestrado [Barros, 96] [Soares, 97].

Nos trabalhos que podem utilizar esta tese como uma referência de partida, podem ser identificados três grupos:

- o primeiro é relativo a trabalhos teóricos complementares. Vários foram identificados ao longo da tese. Como exemplos refiram-se: a necessidade de construção de espaços de estados condensados utilizáveis para verificação e para execução do modelo, a execução distribuída de especificações;
- o segundo refere-se à definição e desenvolvimento de novas ferramentas aplicáveis na análise e execução das RdP-R e RdP-RH. É uma área em que é estritamente necessário ter presente as ferramentas actualmente disponíveis, bem como (as propostas de) normas em preparação no âmbito das organizações internacionais de normalização. Como exemplos

de ferramentas particularmente interessantes para a efectiva aplicação das RdP-R e RdP-RH em situações de engenharia (ou do ensino de engenharia), refiram-se aquelas que permitem a sua execução em plataformas específicas, de entre as quais se mencionam tradutores e bibliotecas para linguagens específicas, nomeadamente C++, VHDL/Verilog e linguagens nativas de controladores ou de dispositivos de lógica programável;

- o terceiro agrupa os trabalhos associados à aplicação das RdP-R e RdP-RH a áreas específicas, como seja a dos edifícios inteligentes e toda a panóplia de sistemas digitais, embebidos e de automação aí presentes. Ao longo da tese já foram identificados vários temas em aberto, pelo que se considera desnecessário proceder a repetições.

O número e diversidade das propostas pendentes é, desta forma, de tal ordem elevado que será necessária uma meditação serena e partilhada para eleger o(s) caminho(s) prioritário(s) a seguir. De qualquer modo, a identificação, quer ao longo da tese, quer nesta secção, das múltiplas linhas complementares em que se justifica trabalho de investigação adicional, tenta contribuir para que venha, realmente, a existir trabalho a jusante deste agora concluído.

## 8. Bibliografia

- [Adamsky & Monteiro, 95] Marian Adamski, João L. Monteiro; 1995; “PLD Implementations of Logic Controllers”; IEEE International Symposium on Industrial Electronics, ISIE’95 - Atenas, Grécia
- [Adamsky & Monteiro, 96] Marian Adamski, João L. Monteiro; 1996; “Declarative Specification of System Independent Logic Controller Programs”; IEEE International Symposium on Industrial Electronics, ISIE’96 - Varsóvia, Polónia
- [Albertos, 92] P. Albertos; 1992; “Fuzzy controllers”; in “Application of Artificial Intelligence in Process Control”; L. Bougart, A. Krijgsman and R. A. Vingerhoeds (eds); Pergamon Press; ISBN 0 08 042017 6
- [Arkin & Paciuk, 95] H. Arkin and M. Paciuk; 1995; “Service systems integration in Intelligent Buildings”; em *Intelligent Buildings I, New Methods and Technologies in Planning and Construction of Intelligent Buildings*, Proceedings of IB/IC Intelligent Buildings Congress’95; Editor A. Lustig; pp. 19-30
- [Atabakhche et al., 86] H. Atabakhche, D. Simonetti Barbalho, R. Valette, M. Courvoisier; 1986; “From Petri net based PLCs to knowledge based control”; Proceedings IECON’86; 1986 International Conference on Industrial Electronics, Control and Instrumentation; Milwaukee, WI, USA, September 29 - October 3, pp. 817-822
- [Baldassari & Bruno 88] Marco Baldassari, Giorgio Bruno; 1988; “An Environment for Object-Oriented Conceptual Programming Based on PROT Nets”; em “Advances in Petri Nets 1988”; Lecture Notes in Computer Science LNCS-340; G. Rozenberg (Ed.); Springer-Verlag; pp. 1-19
- [Barros, 96] João Paulo Barros; 1996; “CpPNeTS: uma Classe de Redes de Petri de Alto-nível - Implementação de um sistema de suporte à sua aplicação e análise”; Tese de Mestrado; Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia

- [Barros, Gomes & Steiger-Garção, 97] João Paulo Barros, Luís Gomes, A. Steiger-Garção; 1997; "Implementation of a non-autonomous high-level Petri net model for reactive real-systems"; 4th IFAC Workshop on Algorithms and Architectures for Real-Time Control; Vilamoura, Portugal; 9 a 11 Abril 1997
- [Barros, Silva & Soares, 93] João Paulo Barros, Luís Silva e Carlos Soares; ano lectivo de 1992/93; "Simulação de controladores especificados através de redes de Petri coloridas e sincronizadas"; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Programação
- [Bastide & Palanque 93] Rémi Bastide, Philippe Palanque; 1993; "Cooperative Objects: a Concurrent Petri Net Based Object-Oriented Language"; Proceedings of the IEEE System Man and Cybernetics 93 "Systems Engineering in the Service of Humans", Le Touquet, 17-20 October 1993
- [Bastide & Palanque, 95] Rémi Bastide, Philippe Palanque; 1995; "A Petri Net Based Environment for the Design of Event-Driven Interfaces"; ATPN'95; 16th International Conference on Application and Theory of Petri Nets; Torino, Italy; 26 a 30 de Junho 1995; em Lecture Notes in Computer Science LNCS 935; Giorgio De Michelis, Michel Diaz (Eds.), pp. 66-83; Springer Verlag; ISBN 3-540-60029-9
- [Battiston & Cindio 93] E. Battiston, F. de Cindio; 1993; "Class Orientation and Inheritance in Modular Algebraic Nets"; Proceedings of IEEE International Conference on Systems, Man and Cybernetics; Le Touquet, France, pp. 717-723
- [Battiston, Cindio & Mauri 88] E. Battiston, F. de Cindio, G. Mauri; 1988; "OBJSA Nets: A Class of High-level Nets having Objects as Domains"; em "Advances in Petri Nets 1988"; Lecture Notes in Computer Science LNCS-340; G. Rozenberg (Ed.); Springer-Verlag; pp. 20-43
- [Becker, 95] Rachel Becker; 1985; "What is an "Intelligent Building" ?"; in Intelligents Buildings I, New Methods and Technologies in Planning and Construction of Intelligent Buildings, Proceedings of IB/IC Intelligent Buildings Congress'95; Editor A. Lustig; "Discussion paper for Workshop A"; pp. 320
- [Beck, 94] Michael von der Beeck; 1994; "A Comparison of Statecharts Variants"; LNCS 863; Formal Techniques in Real-Time and Fault-Tolerant Systems - Third International Symposium; H. Langmaack, W-P. De Roever, J. Vytupil (Eds.); pp. 128-148

- [Bernardinello & De Cindio, 92] Luca Bernardinello, Fiorella De Cindio; 1992; "A Survey of Basic Net Models and Modular Net Classes"; em "Advances in Petri Nets 1992"; Lecture Notes in Computer Science; G. Rozenberg (Ed.); Springer-Verlag
- [Berry & Cosserat, 85] G. Berry, L. Cosserat; 1985; "The ESTEREL Synchronous Programming Language and Its Mathematical Semantics"; LNCS 197; Springer-Verlag
- [Berry & Gonthier, 92] Gérald Berry, Georges Gonthier; 1992; "The Esterel synchronous programming language: design, semantics, implementation"; Science of Computer Programming, vol. 19, n. 2, pp.83-152
- [Berry, 93] Gérard Berry; 1993; "Preemption in Concurrent Systems"; LNCS 761; Proceedings FSTTCS'93; pp. 72-93
- [Berthlot, 83] G. Berthelot; 1983; "Transformation et analyse de réseaux de Petri, applications aux protocoles"; Thèse d'état. Université P. et M. Curie. Paris
- [Berthlot, 85] G. Berthelot; 1986; "Checking properties of nets using transformations"; em "Advances in Petri Nets 86"; LNCS 222; G. Rozenberg (Ed.); Springer-Verlag
- [Billington, 95] Jonathan Billington; 20-28 Novembro 1995; "Introduction to Communication Protocols"; Second International Course on Petri Nets for Latin America; Campina Grande, PB, Brasil
- [Blake-Thomas, 97] Glan Blake Thomas; "The real and hidden cost benefits to the developer, the user and the nation by designing a truly intelligent building"; comunicação oral no 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997 (não incluída nos Proceedings)
- [Branco, 95] Paulo Branco; ano lectivo de 1994/95; "Seguimento de intrusos através de visão por computador"; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Computação Gráfica
- [Bruno et al., 95] G. Bruno, R. Agarwal, A. Castella, M. P. Pescarmona; 1995; "CAB: an Environment for Developing Concurrent Application"; ATPN'95; 16th International Conference on Application and Theory of Petri Nets; Torino, Italy; 26 a 30 de Junho 1995; em Lecture Notes in Computer Science LNCS 935; Giorgio De Michelis, Michel Diaz (Eds.), pp. 141-160; Springer Verlag; ISBN 3-540-60029-9

- [Bucci, Campanai & Nesi, 95] Giacomo Bucci, Maurizio Campanai, Paolo Nesi; 1995; "Tools for Specifying Real-Time Systems"; *Real-Time Systems*, 8; Kluwer Academic Publishers; pp. 117-172
- [Buchholz, 94] Peter Buchholz; 1994; "Hierarchical High Level Petri Nets for Complex System Analysis"; *ATPN'94*; 15th International Conference on Application and Theory of Petri Nets; Zaragoza, Spain; June 20-24, 1995; em *Lecture Notes in Computer Science LNCS 815*; Robert Valette (Ed.), Springer Verlag; pp. 119-138
- [Buchs & Guelfi 91] D. Buchs, N. Guelfi; 1991; "CO-OPN: A Concurrent Object Oriented Petri Net Approach"; *Proceedings of the 12th International Conference on Application and Theory of Petri Nets*; Gjern, Denmark
- [Bugarín & Barro, 94] Alberto J. Bugarín, Senén Barro; 1994; "Fuzzy Reasoning Supported by Petri Nets"; *IEEE Transactions on Fuzzy Systems*; vol. 2, no. 2, pp. 135-150
- [Calvez, 93] Jean Paule Calvez; 1993; "Embedded Real-Time Systems"; John Wiley & Sons; ISBN 0-471-93563-8
- [Camarinha-Matos & Afsarmanesh, 95] Luís Camarinha-Matos, H. Afsarmanesh (eds.); 1995; "Balanced Automation Systems: architectures and design methods"; Chapman & Hall
- [Camarinha-Matos & Afsarmanesh, 96] Luís Camarinha-Matos, H. Afsarmanesh (eds.); 1996; "Balanced Automation Systems II: implementation challenges for anthropocentric manufacturing"; Chapman & Hall
- [Cao & Sanderson, 93] T. Cao, A. C. Sanderson; 1993; "Variable Reasoning and Analysis about Uncertainty with Fuzzy Petri Nets"; 14th International Conference on Application and Theory of Petri Nets; Chicago; USA; in *LNCS-691*, Marco Ajmone Marsan (Ed.); Springer-Verlag; pp. 126-145
- [Cardoso & Valette, 97] Janette Cardoso, Robert Valette; 1997; "Redes de Petri"; Editora da UFSC, Florianópolis, Brasil
- [Chang, Kalavade & Lee, 96] W.-T- Chang, A. Kalavade, E. A. Lee; 1996; "Effective Heterogenous Design and Co-simulation"; in "Hardware/Software Co-Design", Giovanni De Micheli, Mariagiovanna Sami (Eds.); Kluwer Academic Publishers; pp. 187-212

- [Chen, Ke & Chang, 90] Shyi-Ming Chen, Jyh-Sheng Ke and Jin-Fu Chang; 1990; "Knowledge Representation Using Fuzzy Petri Nets"; IEEE Transactions on Knowledge and Data Engineering, vol. 2, no. 3, pp. 311-319
- [Christensen & Hansen, 93] Soren Christensen, Niels Damgaard Hansen; 1993; "Coloured Petri Nets Extended with Place Capacities, Test Arcs and Inhibitor Arcs"; ATPN'93; 14th International Conference on Application and Theory of Petri Nets; Chicago, Illinois, USA; June, 1993; em Lecture Notes in Computer Science LNCS 691; Marco Ajmone Marsan (Ed.), Springer Verlag; pp. 186-205
- [Christensen & Hansen, 94] Soren Christensen, Niels Damgaard Hansen; 1994; "Coloured Petri Nets Extended with Channels for Synchronous Communication"; ATPN'94; 15th International Conference on Application and Theory of Petri Nets; Zaragoza, Spain; June 20-24, 1995; em Lecture Notes in Computer Science LNCS 815; Robert Valette (Ed.), Springer Verlag; pp. 159-178
- [Christensen, sd] Soren Christensen; s.d.; "Coloured Petri Nets: Theory, Tools and Practice"; relatório interno, Computer Science Department, Aarhus University; obtido por ftp
- [Chun & Bien, 93] Myung-Geun Chun, Zeungnam Bien; 1993; "Fuzzy Petri Net Representation and Reasoning Methods for Rule-Based Decision Making Systems"; IEICE Trans. Fundamentals, vol. E-76-A, no. 6, pp. 974-982
- [Clare, 73] Christopher R. Clare; 1973; "Designing logic systems using state machines"; McGraw-Hill
- [Costa et al., 95] Alessandra Costa, Alessandro de Gloria, Paolo Faraboschi, Andrea Pagni, Gianguido Rizzotto; 1995; "Hardware Solutions for Fuzzy Control"; Proceedings of the IEEE, vol. 83, no. 3, pp. 422-434
- [David & Alla, 92] René David, Hassane Alla; 1992; "Petri nets and Grafcet - Tools for modelling discrete event systems"; Prentice Hall; ISBN 0-13-327537-X
- [David, 91] René David; 1991; "Modeling of dynamic systems by Petri nets"; in Proceedings of the ECC91 European Control Conference, Grenoble, France, July 2-5; pp. 136-147
- [DEGW, Arup & Northcroft, 96] DEGW International Consulting Ltd, Ove Arup & Partners, Northcroft; March 1996; "Intelligent Buildings in South East Asia - Executive Summary"; publicado pelo "European Intelligent Building Group"; ISBN 1 901195 00 7

- [DEGW & Teknibank, 95] DEGW (London), Teknibank (Milan), EIBG; 1995; "The Intelligent Building in Europe - executive summary"; em *Intelligent Buildings I, New Methods and Technologies in Planning and Construction of Intelligent Buildings*, Proceedings of IB/IC Intelligent Buildings Congress'95; Editor A. Lustig; pp. 186-209
- [Desel & Merceron, 89] Jörg Desel, Agathe Merceron; 1989; "P/T-systems as Abstractions of C/E-systems"; em LNCS 424, *Advances in Petri Nets 1989*; G. Rozenberg (Ed.); Springer-Verlag; pp. 105-127;
- [Fatrónica, 93] *Fatrónica*; 1993; "Manual de Programação PLCs *Fatrónica*"; 5ª edição
- [Fehling, 93] Rainer Fehling; 1993; "A Concept of Hierarchical Petri Nets with Building Blocks"; em "Advances in Petri Nets 1993"; Grzegorz Rozenberg (Ed.); Springer-Verlag; pp. 148-168
- [Fernandes, 94] João Miguel Fernandes; 1994; "Redes de Petri e VHDL na Especificação de Controladores Paralelos"; Tese de Mestrado; Universidade do Minho, Escola de Engenharia, Departamento de Informática
- [Fernandes & Proença, 94] João Miguel Fernandes, Alberto José Proença; 1994; "Redes de Petri na Especificação e Validação de Controladores Paralelos"; 1º Encontro Nacional do Colégio de Engenharia Electrotécnica; Ordem dos Engenheiros; Lisboa; Maio 1994
- [Fernandes et al., 95] João M. Fernandes, Alberto J. Proença, Marian Adamski; 1995; "VHDL Generation from Petri Net Parallel Controller Specifications"; VHDL'95
- [Gama & Correia, 92] Luís Manuel Dias Gama e Nuno Miguel Bravo Correia; ano lectivo de 1991/92; "Análise da especificação de controladores através de redes de Petri coloridas e sincronizadas"; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Sistemas de Exploração
- [Genrich, 86] H. J. Genrich; 1986; "Predicate/Transition Nets"; LNCS 254, *Advances in Petri Nets 1986 Part I*, W. Brauer, W. Reisig, G. Rozenberg (Eds.); Springer-Verlag; pp. 207-247; incluído também em [Jensen & Rozenberg, 91]
- [Godefroid, 96] Patrice Godefroid; 1996; "Partial-Order Methods for the Verification of Concurrent Systems - An Approach to the State-Explosion Problem"; LNCS 1032; Springer; ISBN 3-540-60761-7
- [Gomes et al., 93] Luís Gomes, A. Steiger-Garção, Luís Gama, Nuno Correia; 1993; "Programação de controladores utilizando redes de Petri"; 4ª Jornadas Nacionais de Projecto,

- Planeamento e Produção Assistidos por Computador; Ordem dos Engenheiros; Lisboa; 17 a 19 Maio 93, pp. 231-238; e ainda Revista "Ingenium" nº 72 Julho/Agosto 1993 pp. 43-52
- [Gomes et al., 94a] Luís Gomes, A. Steiger-Garção, Diogo Assunção, Nuno Matamouros; 1994; "Monitoração de edificios sensorialmente bem equipados"; 1º Encontro Nacional do Colégio de Engenharia Electrotécnica; Ordem dos Engenheiros; FIL; Lisboa; 12 a 13 Maio 94; pp. 43-48
- [Gomes et al., 94b] Luís Gomes, A. Steiger-Garção, João P. Barros, P. Próspero Luís; 1994; "Simulação de controladores especificados através de redes de Petri"; 1º Encontro Português de Controlo Automático - CONTROLO'94; IST; Lisboa; 14 a 16 Setembro 94; pp. II-225-230
- [Gomes et al., 95] Luís Gomes, João Paulo Barros, Carlos Soares, A. Steiger-Garção; 1995; "Monitoração de arquitecturas sensoriais distribuídas"; 5ª Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador; Ordem dos Engenheiros; Guimarães; 30 de Maio a 1 de Junho 95; pp. 223-228
- [Gomes et al., 97] Luís Gomes, Anikó Costa, Carlos Soares, João-Paulo Barros, Adolfo Steiger-Garção; 1997; "Campus-Guard: a domot targeted for integrated building monitoring and control "; in New Methods and Technologies in Planning and Construction of Intelligent Buildings II, Proceedings of the 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997, A. Lustig (Ed.), ISSN 0793-2448, pp 53-64
- [Gomes & Steiger-Garção, 92] Luís F.S. Gomes, A. Steiger Garção; 1992; "Especificação e realização de controladores utilizando redes de Petri coloridas e sincronizadas integrando lógica imprecisa"; Workshop Ibero-americano de Sistemas Autónomos em Robótica e CIM; Lisboa; 2 a 4 de Novembro 1992; Publicado em "Revista Robótica e Automatização", nº 10, Nov 1992, pp 31-38
- [Gomes & Steiger-Garção, 94] Luís Gomes, Adolfo Steiger-Garção; 1994; "Towards different buildings"; Conferencia Internacional sobre el Control Automático en el Ahorro de Energia; La Habana - Cuba; 21 a 23 de Fevereiro 1994;
- [Gomes & Steiger-Garção, 95a] Luís Gomes, Adolfo Steiger-Garção; 1995; "Petri net based Programmable Fuzzy Controller targeted for distributed control environments"; FUZZ-IEEE/IFES'95; The International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and the Second International Fuzzy Engineering Symposium; Yokohama, Japan; 20 a 24 de Março 1995; pp. 1427-1434

- [Gomes & Steiger-Garção, 95b] Luís Gomes, A. Steiger-Garção; 1995; "Integração de diferentes paradigmas de controlo em controladores industriais de baixo custo"; 5ª Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador; Ordem dos Engenheiros; Guimarães; 30 de Maio a 1 de Junho 95; pp. 103-110
- [Gomes & Steiger-Garção, 95c] Luís Gomes, Adolfo Steiger-Garção; 1995; "Fuzzy Petri net model for real-time control"; AARTC'95; Preprints of the 3rd IFAC/IFIP Workshop on Algorithms and Architectures for Real-Time Control; Ostend, Belgium; 31 de Maio a 2 de Junho 1995; pp. 467-472
- [Gomes & Steiger-Garção, 95d] Luís Gomes, Adolfo Steiger-Garção; 1995; "Programmable controller design based on a synchronized colored Petri net model and integrating fuzzy reasoning"; ATPN'95; 16th International Conference on Application and Theory of Petri Nets; Torino, Italy; 26 a 30 de Junho 1995; em Lecture Notes in Computer Science LNCS 935; Giorgio De Michelis, Michel Diaz (Eds.), pp. 218-237; Springer Verlag; ISBN 3-540-60029-9
- [Gomes & Steiger-Garção, 95e] Luís Gomes, Adolfo Steiger-Garção; 1995; "Fuzzy Petri net controller"; IFSA'95; 6th International Fuzzy System Association World Congress; São Paulo, Brasil; 22 a 28 de Julho 1995; pp. 337-340
- [Gomes & Steiger-Garção, 96a] Luís Gomes, Adolfo Steiger-Garção; 1996; "Domots are coming! Or How to manage building automation in a balanced way?"; BASYS'96; 2<sup>nd</sup> IEEE/ECLA/IFIP International Conference on Architectures and Design Methods for Balanced Automation Systems; Lisboa, Portugal; 17-19 Junho 1996
- [Gomes & Steiger-Garção, 96b] Luís Gomes, Adolfo Steiger-Garção; 1996; "Towards the implementation of conflict resolution on a non-autonomous High-level Petri net model"; 1<sup>st</sup> Int. Workshop on Manufacturing and Petri Nets, integrado em ATPN'96 - 17th International Conference on Application and Theory of Petri Nets; Osaka, Japão; 24 a 28 de Junho 1996
- [Gomes & Steiger-Garção, 96c] Luís Gomes, Adolfo Steiger-Garção; 1996; "Petri net based hierarchical state machine model for reactive real-time systems"; CONTROLO'96; Porto, Portugal; 11 a 13 Setembro 1996
- [Gomes, 91a] Luís F.S. Gomes; Novembro 91; "Especificação de sistemas digitais - Trabalho de síntese"; Provas de Aptidão Pedagógica e Capacidade Científica; Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

- [Gomes, 91b] Luís F.S. Gomes; Novembro 91; "Lógica Programável - Relatório de uma Aula Prática"; Provas de Aptidão Pedagógica e Capacidade Científica; Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
- [Gomes, 96] Luís Gomes; Setembro 96; "Redes de Petri: aplicação a sistemas de automação"; em "Notas da I Escola Regional de Informática", pp 5-43, UNESP, Rio Preto, São Paulo, Brasil
- [Gomes, 97] Luís Gomes; 1997; "Fuzzy modelling for reactive real-time systems control using Reactive Petri Nets"; capítulo submetido para inclusão em "Fuzziness in Petri Nets"; J. Cardoso, H. Scarpelli (Eds.), Springer; publicação prevista para 1997
- [Gonçalves, 95] Avelino Silva Gonçalves; Fevereiro a Junho 1995; "Système de contrôle d'accès aux édifices"; Relatório de Projecto Final de Licenciatura opção em "Automatique-Robotique-Systemes"; programa ERASMUS; instituição de origem: Ecole Nationale d'Ingénieurs de Tarbes - França; instituição de acolhimento: UNINOVA-CRI
- [Grafcet, 92] ADEPA - "Agence Nationale pour le Développement de la Productique Appliqué à l'Industrie", AFCET - "Association Française des Sciences et Technologies de l'Information et des Systemes", Noël Bouteille, Paul Brard, Gérard Colombari, Norberto Cotaina, Daniel Richet; 1992; Le Grafcet; Cépaduès-Éditions; ISBN 2-85428-307-4
- [Hack, 75] M. Hack; December 1975; "Decidability Questions for Petri Nets"; Ph. D. Dissertation, Dep. Of Electrical Engineering, Massachusetts Institute of Technology
- [Haddad, 90] S. Haddad; 1990; "A Reduction Theory for Coloured Nets"; em "Advances in Petri Nets 1989", Lecture Notes in Computer Science, LNCS-424; G. Rozenberg (ed.); Springer; pp. 209-235; incluído também em [Jensen & Rozenberg, 91]
- [Halbwachs, 93] Nicolas Halbwachs; 1993; "Synchronous programming of reactive systems"; Kluwer Academic Publishers; ISBN 0-7923-9311-2
- [Han, 97] Chia Yung Han; 1997; "An OOP-Based Intelligent Building System"; in New Methods and Technologies in Planning and Construction of Intelligent Buildings II, Proceedings of the 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997, A. Lustig (Ed.), ISSN 0793-2448, pp 18-29
- [Harel & Naamad, 96] David Harel, Amnon Naamad; 1996; "The STATEMATE Semantics of Statecharts"; Relatório técnico disponível em <ftp://ftp.wisdom.weizmann.ac.il/pub/CSreports/reps95/95-31.ps.z> correspondente à revisão de "The Semantics od Statecharts"; Technical report, I-Logic, Inc., 1989 e 1991

- [Harel & Pnueli, 85] D. Harel, A. Pnueli; January 1985; "On the Development of Reactive Systems"; NATO ASI Series F13; Logics and Models of Concurrent Systems; K. R. Apt (Ed.); Springer, pp. 477-498
- [Harel, 87] David Harel; 1987; "Statecharts: a visual formalism for complex systems"; Science of Computer Programming, 8, pp. 231-274
- [Harel, 88] David Harel; 1988; "On visual formalisms"; Communications of the ACM, May 1988, Volume 31, Number 5, pp. 514-530
- [He & Lee 91] Xudong He, John A. N. Lee; August 1991; "A Methodology for Constructing Predicate Transition Net Specifications"; Software - Practice and Experience, vol. 21(8), pp. 845-875; John Wiley & Sons, Ltd
- [He 96] Xudong He; 1996; "A Formal Definition of Hierarchical Predicate Transition Nets"; ATPN'96; 17th International Conference on Application and Theory of Petri Nets; Osaka, Japão; Junho 1996; em Lecture Notes in Computer Science LNCS 1095; Jonathan Billington, Wolfgang Reisig (Eds.), pp. 212-229; Springer Verlag; ISBN 3-540-61363-3
- [HPAC, 96] HPAC- Heating/Piping/Air Conditioning; February 1996; "Protocol roundup"
- [Huber et al., 86] P. Huber, A. M. Jensen, L. O. Jepsen and K. Jensen; 1986; "Reachability Trees for High-level Petri Nets"; Theoretical Computer Science 45; Elsevier Science Publishers B. V.; pp. 261-292; também incluído em [Jensen & Rozenberg, 91]
- [Huber, Jensen & Shapiro 90] P. Huber, K. Jensen, R.M. Shapiro; 1990; "Hierarchies in Coloured Petri Nets"; em "Advances in Petri Nets 1990", Lecture Notes in Computer Science LNCS 483, G. Rozenberg (Ed.); pp. 313-341; também incluído em [Jensen & Rozenberg, 91]
- [IBI, 87] IBI - Intelligent Buildings Institute; 1987; "Intelligent Building Definition - Guideline"; Public report; Washington D.C. USA
- [IEC, 92] International Electrotechnical Commission; IEC 1131-1: Programmable Controllers - Part 3: Programming Languages
- [ISEnto, 94a] Luís Gomes; Abril 1994; "ISEnto / Integração de Serviços em Edifícios sensorialmente infra-estruturados - Proposta de projecto"; Relatório interno; ISENTO-RT-1-94;

- [ISEnto, 94b] Luís Gomes; Junho 1994; "ISEnto / Integração de Serviços em Edifícios sensorialmente infra-estruturados - Objectivos e descrição de actividades"; Relatório interno; ISENTO-RT-2-94;
- [ISEnto, 94c] Luís Gomes; Julho 1994; "ISEnto / Integração de Serviços em Edifícios sensorialmente infra-estruturados - Especificação preliminar - documento de trabalho"; Relatório interno; ISENTO-RT-3-94;
- [ISEnto, 94d] Luís Gomes; Dezembro 1994; "Os controladores da Patrónica no projecto ISEnto - versão preliminar"; Relatório interno; ISENTO-RT-4-94;
- [ISEnto, 95a] Luís Gomes, Carlos Soares, João Paulo Barros, Anikó Costa; Outubro 1995; "Os controladores da Patrónica no projecto ISEnto - versão 1"; Relatório interno; ISENTO-RT-1-95;
- [ISEnto, 95b] Luís Gomes, João Paulo Barros, Carlos Soares; Outubro 1995; "Modelo de dados para projecto ISEnto"; Relatório interno; ISENTO-RT-2-95;
- [ISEnto, 95c] João Paulo Barros, Luís Gomes, Carlos Soares, Anikó Costa; Novembro 1995; "Utilitário de teste de servidores: MONITOR versão 1 - Desenvolvimento e utilização - "; Relatório interno; ISENTO-RT-3-95;
- [ISEnto, 96a] João Paulo Barros, Luís Gomes, Carlos Soares, Anikó Costa; Abril 1996; "Protótipo de servidor de nomes / versão 1 - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-1-96;
- [ISEnto, 96b] João Paulo Barros, Luís Gomes, Carlos Soares, Anikó Costa; Abril 1996; "Protótipo de servidor de linha série / versão 1 - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-2-96;
- [ISEnto, 96c] João Paulo Barros, Carlos Soares, Luís Gomes, Anikó Costa; Abril 1996; "Protótipo de servidor de sensores e actuadores / versão 1 - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-3-96;
- [ISEnto, 96d] Carlos Soares, João Paulo Barros, Luís Gomes, Anikó Costa; Abril 1996; "Protótipo de servidor de câmaras / versão 1 - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-4-96;
- [ISEnto, 96e] Anikó Costa, Luís Gomes, Carlos Soares, João Paulo Barros; Abril 1996; "Protótipo de servidor de central telefónica / versão 1 - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-5-96;

- [ISEnto, 96f] Anikó Costa, Luís Gomes, Carlos Soares, João Paulo Barros; Junho 1996; "Interface de utilizador "Campus Guard - versão 1" - desenvolvimento e utilização"; Relatório interno; ISENTO-RT-6-96;
- [Jafari & Boucher, 94] Mohsen A. Jafari, Thomas O. Boucher; 1994; "A rule-based system for generating a ladder logic control program from a high-level systems model"; Journal of Intelligent Manufacturing, 5, pp. 103-120
- [Jager, 95] René Jager; 1995; "Fuzzy Logic in Control"; Thesis Technische Universiteit Delft
- [Jensen & Rozenberg, 91] K. Jensen, G. Rozenberg (Eds.); 1991; "High-level Petri nets - Theory and Application"; Springer-Verlag; ISBN 3-540-54125-X
- [Jensen, 86] K. Jensen; 1986; "Coloured Petri Nets"; LNCS 254, Advances in Petri Nets 1986 Part I, W. Brauer, W. Reisig, G. Rozenberg (Eds.); Springer-Verlag; pp. 248-299; incluído também em [Jensen & Rozenberg, 91]
- [Jensen, 90] K. Jensen; 1990; "Coloured Petri Nets: A High Level Language for System Design and Analysis"; LNCS 483, Advances in Petri Nets 1990, G. Rozenberg (ed.); Springer, pp.342-416; incluído também em [Jensen & Rozenberg, 91]
- [Jensen, 92] Kurt Jensen; 1992; "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 1"; Springer-Verlag; ISBN 3-540-55597-8
- [Jensen, 95] Kurt Jensen; 1995; "Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 2"; Springer-Verlag; ISBN 3-540-58276-2
- [Kesten & Pnueli, 92] Y. Kesten, A. Pnueli; 1992; "Timed and Hibrid Statecharts and Their Textual Representation"; LNCS 571, Springer, pp.591-620
- [Klir & Yuan, 96] George J. Klir, Bo Yuan (Editors); 1996; "Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems - Selected Papers by Lotfi A. Zadeh"; Advances in Fuzzy Systems - Applications and Theory Vol.6; World Scientific
- [Kóczy, 95] László Kóczy; 1995; "Algorithms and models of fuzzy reasoning and control"; Tutorial; VI International Fuzzy Systems Association World Congress; São Paulo, Brasil
- [Konar & Mandal, 96] Amit Konar, Ajit K. Mandal; 1996; "Uncertainty Management in Expert Systems Using Fuzzy Petri Nets"; IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 1, pp. 96-105

- [Lakos & Christensen, 94] Charles Lakos, Soren Christensen; 1994; "A General Systematic Approach to Arc Extensions for Coloured Petri Nets"; ATPN'94; 15th International Conference on Application and Theory of Petri Nets; Zaragoza, Spain; June 20-24, 1995; em Lecture Notes in Computer Science LNCS 815; Robert Valette (Ed.), Springer Verlag; pp. 338-355
- [Lakos & Keen, 94] Charles Lakos, C. Keen; 1994; "LOOPN++: A New Language for Object-Oriented Petri Nets"; Proceedings of Modelling and Simulation (European Simulation Multiconference) Barcelona, Espanha; pp. 369-374
- [Lakos, 95] Charles Lakos; 1995; "From Coloured Petri Nets to Object Petri Nets"; ATPN'95; 16th International Conference on Application and Theory of Petri Nets; Torino, Italy; 26 a 30 de Junho 1995; em Lecture Notes in Computer Science LNCS 935; Giorgio De Michelis, Michel Diaz (Eds.), pp. 178-297; Springer Verlag; ISBN 3-540-60029-9
- [Lakos, 96] Charles Lakos; 1996; "The Consistent Use of Names and Polymorphism in the Definition of Object Petri Nets"; ATPN'96; 17th International Conference on Application and Theory of Petri Nets; Osaka, Japão; 24-28 Junho 1996; em Lecture Notes in Computer Science LNCS 1091; Jonathan Billington, Wolfgang Reisig (Eds.), pp. 380-399; Springer Verlag; ISBN 3-540-61363-3
- [Laplante, 93] Phillip Laplante; 1993; "Real-Time Systems Design and Analysis, an Engineer's Handbook"; IEEE Press; ISBN 0-7803-0402-0
- [Lee, 90] Chuen Chien Lee; 1990; "Fuzzy Logic in Control Systems: Fuzzy Logic Controller"; IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 2, Part I - pp. 404-418, Part II - pp. 419-435
- [Li, 94] Liwu Li; 1994; "High-Level Petri Net Model of Logic Program with Negation"; IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 3, pp. 382-395
- [Looney,88] Carl G. Looney; 1988; "Fuzzy Petri Nets for Rule-Based Decisionmaking"; IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-18, no. 1, pp. 178-183
- [Lucas, 93] Paul Jay Lucas; 1993; "An object-oriented language system for implementing concurrent, hierarchical, finite state machines"; MSc Thesis; Graduate College of the University of Illinois at Urbana-Champaign
- [Luís & Ventura, 92] Pedro Jorge Próspero Luís e Paula Cristina Ventura; ano lectivo de 1991/92; "Editor gráfico de redes de Petri e esquemas lógicos"; Relatório de Projecto Final de

Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Programação

- [Luo & Kay, 89] Ren C. Luo, Michael Kay; 1989; "Multisensor integration and fusion in intelligent systems"; IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-19, nº 5; pp. 901-931
- [Lustig, 95] A. Lustig (Ed.); 1995; "New Methods and Technologies in Planning and Construction of Intelligent Buildings I", Proceedings of the 1<sup>st</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel
- [Lustig, 97] A. Lustig (Ed.); 1997; "New Methods and Technologies in Planning and Construction of Intelligent Buildings II", Proceedings of the 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997, ISSN 0793-2448
- [Machado, 96] Ricardo Machado; 1996; "Hierarquia em Redes de Petri Orientadas por Objectos na Especificação de Sistemas Digitais"; Tese de Mestrado; Universidade do Minho, Escola de Engenharia, Departamento de Informática
- [Mamdani, 74] E. H. Mamdani; 1974; "Applications of fuzzy algorithm for simple dynamic plant"; Proceedings IEE 121 (12), 1585-1588
- [Mamdani & Assilian, 75] E. H. Mamdani, S. Assilian; 1975; "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller"; International Journal of Man-Machine Studies, 7:1-13
- [Marsan et al., 84] M. Ajmone Marsan, G. Balbo, G. Conte; Maio 1984; "A class of generalised stochastic Petri nets for the performance analysis of multiprocessor systems"; ACM Transactions on Computer Systems, 2(1)
- [Martinez et al., 88] J. Martinez, P. R. Muro, M. Silva, S. F. Smith, J. L. Villarrol; 1988; "Merging artificial intelligence techniques and Petri nets for real time scheduling and control of production systems"; 12<sup>th</sup> IMACS World Congress on Scientific Computation; July 18-22, 1988; Paris, France
- [Matamouros & Assunção, 93] Nuno Matamouros e Diogo Assunção; ano lectivo de 1992/93; "Monitoração/acção em edifícios sensorialmente bem equipados"; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Sistemas Distribuídos

- [McKinley, 88] A. D. McKinley; 1988; “Modern Building Services - More than Computerized HVAC”; em Intelligent Buildings, ASHRAE Technical Data Bulletin, pp. 10-21
- [Miyagi, 96] Paulo Eigi Miyagi; 1996; “Controle Programável - fundamentos do controle de sistemas a eventos discretos”; Editora Edgard Blücher Ltda, S. Paulo, Brasil
- [Morgado et al., 94] Carmen Morgado, Miguel Boavida, João Paulo Santos e Pedro Serpa; ano lectivo de 1993/94; “Monitoração e controlo integrado em edifícios”; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Programação
- [Mota & Mendes, 94] João Mota e João Mendes; ano lectivo de 1993/94; “Realidade virtual virtualmente real”; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Computação Gráfica
- [Moura-Pires, 93] Fernando Moura-Pires; 1993; “Aprendizagem por Indução Empírica - Integração em arquitecturas de sistemas sensoriais”; Tese de Doutoramento; Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia
- [Murata & Zhang, 88] Tadao Murata, Du Zhang; 1988; “A predicate/transition net model for parallel interpretation of logic programs”; IEEE Transactions Software Eng., vol. 14, no. 4, pp. 481-497
- [Murata, 89] Tadao Murata; Abril 1989; “Petri nets: Properties, Analysis and Applications”; Proceedings of the IEEE, vol. 77, nº 4, pp. 541-580
- [Murata, Subrahmanian & Wakayama, 91] Tadao Murata, V. S. Subrahmanian, Toshiro Wakayama; 1991; “A Petri Net Model for Reasoning in the Presence of Inconsistency”; IEEE Transactions on Knowledge and Data Engineering, vol. 3, no. 3, pp. 281-292
- [Natkin, 80] S. Natkin; Junho 1980; “Les réseaux de Petri stochastiques”; Thèse de Doctorat, CNAM, Paris
- [Ohr, 90] Stephan A. Ohr; 1990; “CAE: A Survey of Standards Trends and Tools”; John Wiley & Sons
- [PALASM, 82] National Semiconductor Corporation; 1982; “PAL Databook”
- [PALASM, 90] Advanced Micro Devices; 1990; “PALASM User’s Manual”

- [Pedrycz & Gomide, 94] Witold Pedrycz, Fernando Gomide; 1994; "A Generalized Fuzzy Petri Net Model"; IEEE Transactions on Fuzzy Systems; vol. 2, no. 4, pp. 295-301
- [Pedrycz, 93] Witold Pedrycz; 1993; "Fuzzy Control and Fuzzy Systems"; Second, extended, edition; Research Studies Press Ltd; ISBN 0863801315
- [Peterka & Murata, 89] George Peterka and Tadao Murata; 1989; "Proof procedure and answer extraction in Petri net model of logic programs"; IEEE Transactions Software Eng., vol. 15, no. 2, pp. 209-217
- [Peterson, 77] J. Peterson; 1977; "Petri Nets"; Computing Surveys, Volume 9, Number 3; September 1977; pp 223-252
- [Peterson, 81] James L. Peterson; 1981;"Petri Net Theory and the Modeling of Systems"; Prentice-Hall, Inc.
- [Pimenta-Rodrigues & Araújo, 88] Victor Pimenta Rodrigues, Mário Seia de Araújo; 1988; "Projecto de sistemas digitais"; Editorial Presença
- [Prata & Nunes, 93] E. Prata, A. Nunes; 1993; "PLC1B - Autómato programável de última geração desenhado e fabricado em Portugal"; 4ª Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador; Ordem dos Engenheiros; Fundação Calouste Gulbenkian, Lisboa; pp 239-246
- [Rasmussen & Singh, 96] Jens Linneberg Rasmussen, Mejar Singh; Junho 1996; "Designing a Security System by Means of Coloured Petri Nets"; ATPN'96; 17th International Conference on Application and Theory of Petri Nets; Osaka, Japão; 24 a 28 de Junho 1996; em Lecture Notes in Computer Science LNCS 1091; Jonathan Billington, Wolfgang Reisig (Eds.), pp. 400-419; Springer Verlag; ISBN 3-540-61363-3
- [Recalde, Teruel & Silva, 96] Laura Recalde, Enrique Teruel, Manuel Silva; 1996; "{SC}\*ECS: A Class of Modular and Hierarchical Cooperating Systems"; ATPN'96; 17th International Conference on Application and Theory of Petri Nets; Osaka, Japão; 24 a 28 de Junho 1996; em Lecture Notes in Computer Science LNCS 1091; Jonathan Billington, Wolfgang Reisig (Eds.), pp. 440-459; Springer Verlag; ISBN 3-540-61363-3
- [Reisig, 87] Wolfgang Reisig; 1987; "Petri Nets in software engineering"; Lecture Notes in Computer Science, LNCS 255; Springer-Verlag; pp. 63-96
- [Reisig, 82] Wolfgang Reisig; 1982; "Petri Nets - An Introduction"; Springer-Verlag

- [Reisig, 92] Wolfgang Reisig; 1992; "A Primer in Petri Design"; Springer-Verlag; ISBN 3-540-52044-9
- [Rozenberg & Thiagarajan, 86] G. Rozenberg, P. S. Thiagarajan; 1986; "Petri nets: basic notions, structure, behaviour"; in Current Trends in Concurrency, LNCS 224; J.W. de Bakker, W. -P. De Roever, G. Rozenberg (Eds.); Springer-Verlag
- [Schöf, Sonnenschein & Wieting, 95] Stefan Schöf, Michael Sonnenschein, Ralf Wieting; 1995; "Efficient Simulation of THOR Nets"; ATPN'95; 16th International Conference on Application and Theory of Petri Nets; Torino, Italy; 26 a 30 de Junho 1995; em Lecture Notes in Computer Science LNCS 935; Giorgio De Michelis, Michel Diaz (Eds.), pp. 412-431; Springer Verlag; ISBN 3-540-60029-9
- [Seco & Duarte, 93] João Seco e Joana Duarte; ano lectivo de 1992/93; "Obtenção de imagens 3D em ambientes estruturados"; Relatório de Projecto Final de Licenciatura em Engenharia Informática; disciplinas de Projecto de Robótica e Projecto de Computação Gráfica;
- [Shahdad et all, 85] Moe Shahdad, Roger Lipsett, Erich Marschner, Kellye Sheehan, Howard Cohen, Ron Waxman, Dave Ackley; 1985; "VHSIC Hardware Description Language"; IEEE Computer; February 1985; pp. 94-103
- [Shemie, 97] Sam Shemie; 1997; "Communications, Control & Information Technology Systems in Intelligent Buildings"; in New Methods and Technologies in Planning and Construction of Intelligent Buildings II, Proceedings of the 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997, A. Lustig (Ed.), ISSN 0793-2448, pp 65-82
- [Shimura, Lobo & Murata, 95] Teruhiro Shimura, Jorge Lobo, Tadao Murata; 1995; "An Extended Petri Net Model for Normal Logic Programs"; IEEE Transactions on Knowledge and Data Engineering, vol. 7, no. 1, pp. 150-162
- [Sibertin-Blanc, 93] C. Sibertin-Blanc; 1993; "A Client-Server Protocol for the Composition of Petri Nets"; ATPN'93; 14th International Conference on Application and Theory of Petri Nets; Chicago, Illinois, USA; June, 1993; em Lecture Notes in Computer Science LNCS 691; Marco Ajmone Marsan (Ed.), Springer Verlag; pp. 377-396
- [Sibertin-Blanc, 94] C. Sibertin-Blanc; 1994; "Cooperative Nets"; ATPN'94; 15th International Conference on Application and Theory of Petri Nets; Zaragoza, Spain; June 20-24, 1995; em Lecture Notes in Computer Science LNCS 815; Robert Valette (Ed.), Springer Verlag; pp. 471-490

- [Sibertin-Blanc, Hameurlain & Touzeau, 95] C. Sibertin-Blanc, N. Hameurlain, P. Touzeau; 1995; "SYROCO: A C++ Implementation of Cooperative Objects"; Proceedings of the 1<sup>st</sup> Workshop on Object-Oriented Programming and Models of Concurrency, Torino, Italy; associado à 16th International Conference on Application and Theory of Petri Nets
- [Silva & Valette, 89] M. Silva, R. Valette; 1989; "Petri nets and Flexible Manufacturing"; in Advances in Petri Nets, 1989; Lecture Notes in Computer Science; vol. 424; G. Rozenberg (ed.); Springer, Berlin; pp 374-417
- [Silva & Velilla, 82] M. Silva, S. Velilla; 1982; "Programmable Logic Controllers and Petri Nets: a comparative study"; IFAC Software for Computer Control, Madrid, Spain; pp. 83-88
- [Silva, 85] M. Silva; 1985; "Las Redes de Petri: en la Automática y la Informática"; Editorial AC, Madrid, ISBN 84 7288 045 1
- [Silva, 89] M. Silva; 1989; "Logical controllers"; IFAC Low Cost Automation, Techniques, Components & Instruments, Applications, Milan, Italy; pp. F-157, F-166
- [Silva, 96] M. Silva; 1996; "Linear Programming, Convex Geometry and P/T Nets"; 2<sup>nd</sup> Advanced Course on Petri nets; Dagstuhl, Alemanha; 7-18, Outubro, 1996
- [Smith, 95] E. Smith; 1995; "Carl Adam Petri and the continuum"; Second International Course on Petri Nets for Latin America; Campina Grande, Brazil; November 20-28/1995
- [Soares, 97] Carlos Soares; 1997; Tese de Mestrado, em preparação; Univ. Nova de Lisboa, Fac. de Ciências e Tecnologia
- [STATEMATE, 90] D. Harel, H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, A. Shtull-Trauring, M. Trakhtenbrot; Abril 1990; "STATEMATE: A Working Environment for the Development of Complex Reactive Systems"; IEEE Trans. On Software Engineering, Vol. 16, No. 4
- [Takagi & Sugeno, 83] T. Takagi, M. Sugeno; 1983; "Derivation of fuzzy control rules from human operator's control actions"; Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, Marseille, France, pp55-60
- [Takagi & Sugeno, 85] T. Takagi, M. Sugeno; 1985; "Fuzzy identification of systems and its applications to modeling and control"; IEEE Transactions SMC 15; pp. 116-132

- [Thiagarajan, 87] P. S. Thiagarajan; 1987; "Elementary net systems"; in Petri nets: central models and their properties, LNCS 254; W. Brauer, W. Reisig and G. Rozenberg (Eds.); Springer-Verlag; pp. 26-59
- [Tomé, 89] José Alberto Baptista Tomé; 1989; "Controlo de Sistemas Digitais"; Editorial Presença - Colecção Informática e Computadores
- [Treseler, 92] Michael Treseler; 1992; "Designing state machine controllers using programmable logic"; Prentice-Hall; ISBN 0-13-202938-3
- [Vale, 95] José Manuel Rodrigues do Vale; Fevereiro a Junho 1995; "Étude de l'élaboration d'un contrôleur à logique floue avec un automate non dédié"; Relatório de Projecto Final de Licenciatura opção em "Automatique-Robotique-Systèmes"; programa ERASMUS; instituição de origem: Ecole Nationale d'Ingénieurs de Tarbes - França; instituição de acolhimento: UNINOVA-CRI
- [Valmari, 91] A. Valmari; 1991; "Stubborn sets for reduced state space generation"; LNCS 483 - Advances in Petri nets; Springer; pp. 491-515
- [Vivian, 95] Lui Man Vai, Vivian; Janeiro a Dezembro de 1995; "Monitoração de edifícios"; Relatório de Projecto Final de Licenciatura; instituição de origem: Universidade de Macau - Faculdade de Ciências e Tecnologia; instituição de acolhimento: UNINOVA-CRI
- [Yager, 84] Ronald R. Yager; Julho/Agosto 1984; "Approximate reasoning as a basis for rule-based expert systems"; IEEE Transactions on Systems, Man and Cybernetics, vol. 14, no. 4, pp. 636-643
- [Yomura, 97] Márcia Cristina Yomura; 1997; "Analysis of building automation systems"; in New Methods and Technologies in Planning and Construction of Intelligent Buildings II, Proceedings of the 2<sup>nd</sup> IB/IC Intelligent Buildings Congress, Tel-Aviv, Israel, 4-6 March 1997, A. Lustig (Ed.), ISSN 0793-2448, pp 30-40
- [Yu, 95] Sheng-Ke Yu; 1995; "Comments on "Knowledge Representation Using Fuzzy Petri Nets""; IEEE Transactions on Knowledge and Data Engineering, vol. 7, no. 1, pp. 190-192
- [Zimmermann, 85] H.-J. Zimmermann; 1985; "Fuzzy Set Theory and its Applications"; Kluwer Academic Publishers
- [Zadeh, 96] Lotfi A. Zadeh; 1996; "Fuzzy Logic = Computing with Words"; IEEE Transactions on Fuzzy Systems; vol. 4, no. 2, pp. 103-111

**Referências Gerais**

[Machado J.P., 90] José Pedro Machado (coordenador); 1990; “Grande Dicionário da Língua Portuguesa”; Sociedade de Língua Portuguesa; EDICLUBE - Edição e Promoção do Livro, Lda.

[Machado J.P., 96] José Pedro Machado; 1996; “O Grande Livro dos Provérbios”; Editorial Notícias

[Parreira & Pinto, 85] Manuela Parreira, J. Manuel de Castro Pinto; 1985; “Prontuário Ortográfico Moderno”; Edições Asa

## Epílogo

Após a conclusão da escrita do texto desta Tese, concluí que era justificado introduzir uma reflexão pessoal, fora dos aspectos técnicos e científicos abordados no seu corpo.

Nessa reflexão, a Carreira Docente Universitária configura-se-me, predominantemente, como uma carreira de dedicação a causas e valores, ou seja, como uma carreira de entrega desinteressada e de paixão. É, assumidamente, uma visão facilmente classificável como ingénua.

Nesta configuração, compará-la com as actividades religiosas de busca e divulgação missionária da Verdade é uma tentação, a que não renuncio. Nesse contexto, o conhecido conto do Monge e da Sopa de Pedra pode, facilmente, ser alvo de uma nova interpretação. Refiro-me, obviamente, ao conto em que um monge quando, no final do dia, tenta encontrar a alimentação (supostamente) de subsistência, recorre a um estratagema de prometer um óptimo manjar a quem esteja na disposição de fornecer “alguns” ingredientes simples e abundantes (pelo menos no início), com base nas propriedades extraordinárias de uma pequena pedra que transporta consigo. A pedra, que o monge no fim de cada refeição recupera para o seu caminho posterior, é, assim, como que a sua garantia de subsistência. O truque, no conto, resulta e permite que o monge vá sobrevivendo...

Sem desejos de tornar o discurso demasiado gongórico, embora a metáfora saia óbvia, a Tese de Doutoramento comporta-se para o Académico como a Pedra para o Monge do conto. É com ela que o recém-Doutor poderá preparar os seus próximos manjares intelectuais, recuperando-a para uso posterior, depois de a ter utilizado na sua função de catalisador noutros trabalhos técnico-científicos.

Deste modo, o presente trabalho constitui-se na pedra que tentarei utilizar no futuro para garantir a minha subsistência (enquanto académico), muito embora não tenha (neste momento) determinado de forma exacta qual o seu valor real (espero, no entanto, melhorar no futuro próximo o grau de conhecimento sobre o assunto).