



Universidade Federal de Pernambuco
Centro de Informática

Tese de Doutorado

**Avaliação de Dependabilidade de Sistemas com
Mecanismos Tolerantes a Falha: Desenvolvimento
de um Método Híbrido Baseado em EDSPN e
Diagrama de Blocos**

Sérgio Murilo Maciel Fernandes

Recife, 27 de Fevereiro de 2007

Universidade Federal de Pernambuco
Centro de Informática

Sérgio Murilo Maciel Fernandes

Avaliação de Dependabilidade de Sistemas com
Mecanismos Tolerantes a Falha:
Desenvolvimento de um Método Híbrido Baseado
em EDSPN e Diagrama de Blocos

Tese apresentada para a obtenção do título de
Doutor em Ciências da Computação pela
Universidade Federal de Pernambuco - Centro de
Informática.

Orientador: Prof. Dr. Paulo Romero Martins Maciel

Recife, 27 de Fevereiro de 2007

Fernandes, Sérgio Murilo Maciel

Avaliação de dependabilidade de sistemas com mecanismos tolerantes a falha: desenvolvimento de um método híbrido baseado em EDSPN e diagrama de blocos / Sérgio Murilo Maciel Fernandes. – Recife: O autor, 2007.

xvi, 230 folhas : il., fig., tab.

Tese (doutorado) – Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2007.

Inclui bibliografia, glossário e anexos.

1. Computação tolerante a falhas. 2. Dependabilidade. 3. Modelos estocásticos. 4. EDSPN (redes de Petri estocástica determinística estendida). I. Título.

004.2

CDD (22.ed.)

MEI2007-025

Quando as coisas vão erradas não penses que todos os teus esforços tenham sido em vão. Talvez tudo tenha sido para melhor. Sorria e experimente outra vez! Se guardares em mente o alto objetivo das tuas aspirações os teus sonhos se realizarão. Tire proveito dos teus erros. Colha experiência das tuas dores. “E, então, um dia tu dirás: Graças a Deus eu ousei experimentar outra vez.”

(autor desconhecido)

*Dedico esta Tese a minha esposa,
aos meus filhos, a minha mãe, ao meu pai
(In- memorian) e em especial ao
Espírito Santo.*

Agradecimentos

Agradeço a minha família pela preocupação, compreensão e colaboração, nos momentos oportunos, durante todo este período de Tese. Várias foram as demonstrações de carinho e solidariedade. Sem vocês seria impossível chegar até o fim. Agradeço ao meu orientador, pessoa a quem muito estimo, a oportunidade de estarmos lado a lado trabalhando de uma forma séria, ética e responsável. Foi muito importante trabalhar com quem gosta do que faz e procura fazer bem feito. Agradeço aos professores que me deram a oportunidade de ser aceito no programa de doutorado e a todo o pessoal do Centro de Informática da UFPE, desde o seu diretor ao funcionário mais humilde, pela acolhida em seu meio. Agradeço ainda aos colegas de doutorado pelos bons momentos e pela prestimosa ajuda em todas as ocasiões. Foi muito bom tê-los conhecido. Agradeço a Universidade Católica de Pernambuco o apoio dado, especialmente, à Pró - Reitoria de Pesquisa e a chefia do Departamento de Estatística e Informática (DEI). Agradeço ainda as demonstrações de solidariedade e amizade dos funcionários e professores do DEI. Enfim, agradeço a todos que ajudaram, ou procuraram ajudar, na realização deste trabalho. Por fim, o meu agradecimento especial, Àquele que mais me ajudou, me inspirou, me acolheu em todos os momentos e que nunca faltou quando Dele eu mais precisava. Agradeço ao Espírito Santo de Deus que nas noites escuras iluminava os meus caminhos com as luzes das estrelas, que nos dias frios me acolhia com o calor do sol, e na ausência de palavras reconfortantes, me enviava o canto dos passarinhos, na presença doce e constante das “lavadeiras”. Que sempre seja feita a Sua vontade.

Resumo

Nos dias atuais, observamos o vertiginoso avanço da tecnologia e a dependência cada vez maior da sociedade nos sistemas de computação. O uso massivo de dispositivos computadorizados, fixos e móveis, dentro de um conceito de computação ubíqua, e a crescente pervasividade das redes de computadores e serviços, têm tornado os sistemas extremamente complexos e dinâmicos. Esta complexidade vem aumentando a cada dia, a medida que os computadores têm se tornado menores, mais baratos e com maior capacidade de processamento. Hoje eles estão presentes não apenas em grande parte dos objetos da vida diária, como aparelhos celulares, *laptops* e *desktops*, como também nos sistemas de telecomunicações, nos meios de transporte, nos equipamentos hospitalares, e na maior parte das atividades.

Enquanto razões econômicas forçam o desenvolvimento de novos sistemas computacionais, com um número cada vez maior de facilidades, razões de qualidade impõem a necessidade de que sejam evitados maus funcionamentos desses sistemas. As conseqüências das paralisações dos sistemas computacionais podem variar desde simples inconveniências, a perda de vidas humanas ou prejuízos materiais, o que motiva o desenvolvimento de metodologias para a avaliação da dependabilidade, ou segurança de funcionamento, desses sistemas.

Devido ao comportamento aleatório de grande parte das falhas, técnicas de modelagem de dependabilidade por meio de avaliação analítica ou simulação estocástica têm provado ser uma solução útil e versátil em todas as fases do ciclo de vida de um sistema, desde a fase de projeto, na escolha da solução que melhor satisfaça aos requisitos de dependabilidade propostos, até a fase operacional, na detecção de gargalos que impeçam os sistemas de atingir tais requisitos. Esta Tese propõe o desenvolvimento de uma metodologia que possibilite a modelagem, o refinamento e a avaliação de sistemas dependáveis com a utilização de mecanismos de tolerância a falhas, através de um método híbrido baseado em redes de Petri estocásticas determinísticas estendidas (EDSPN) e diagramas de blocos, de um modo flexível, expansível e passível de automação.

A metodologia proposta é executada em 5 diferentes níveis hierárquicos. Os dois primeiros níveis hierárquicos lidam com os diagramas de blocos e com os mecanismos de tolerância a falhas a serem introduzidos. O terceiro nível hierárquico, formado por redes de Petri de alto nível, define como será a interligação das redes de Petri, que representa a configuração dos blocos no diagrama de sistema dependável, na configuração final. No quarto nível hierárquico, o comportamento de cada bloco é modelado por meio de EDSPN, o qual gera expressões numéricas ou analíticas dos atributos de confiabilidade, disponibilidade e segurança. Finalmente, as expressões obtidas são utilizadas em um modelo dependável e parametrizado (MDP), de acordo com a configuração definida no terceiro nível, para obtenção das estimativas de dependabilidade do sistema como um todo. A metodologia proposta, além de ser passível de automação, objetiva ocultar a complexidade matemática envolvida e reduzir a possibilidade de explosão de estados.

Para tornar os modelos EDSPN e MDP mais eficientes, uma biblioteca de modelos foi criada. Um mesmo modelo, com o auxílio de diferentes parâmetros de configuração, pode assumir diferentes mecanismos de tolerância a falhas, o que torna a metodologia flexível, assim como, um mesmo modelo, com o auxílio de parâmetros estruturais, pode assumir diferentes níveis de redundância em um mesmo mecanismo tolerante a falhas, o que torna a metodologia expansível.

Palavras-chave: tolerância a falhas, dependabilidade, modelos estocásticos, EDSPN.

Abstract

In current days, we observe a vertiginous advance of technology and a larger dependence of the society on computational systems. A massive use of computerized devices, fixed and mobile, inside a concept of ubiquitous computation, and the increasing pervasiveness of computers and services networks, have been turning systems extremely complex and dynamic. This complexity is increasing every day, as computers become smaller, cheaper and with larger processing capacity. Today they are not only largely present in objects of the daily life, like cellular devices, laptops and desktops, as well as in telecommunication systems, in the means of transportation, in the hospital equipment, and in major part of activities.

While economic reasons force the development of new computational systems, with an every time larger number of facilities, quality reasons impose the need for avoiding malfunction of these systems. The consequences of computational systems failure can vary since simple inconveniences, to the loss of human lives or material damage, what motivates the development of methodologies for dependability evaluation of these systems.

Due to the stochastic behavior of great part of the faults, techniques of dependability modeling by means of stochastic analytic evaluation or simulation, have been proving be a useful and versatile solution in all phases of life cycle of a system, from the project phase, in the choice of a better solution that satisfies the proposed dependability requirements, until the operational phase, in the detection of bottlenecks that prevent the systems of reaching such requirements. This thesis proposes the development of a methodology that enables modeling, refinement and evaluation of dependable systems with fault tolerance mechanisms utilization, through a hybrid method based on extended deterministic and stochastic Petri nets (EDSPN) and block diagrams, in a flexible and expandable way, that allows automation.

The proposed methodology is implemented in 5 different hierarchical levels. The first two levels work with block diagrams and fault tolerant mechanisms to be introduced. The third level, formed by high level Petri nets, defines the interconnection of the Petri nets, which represents the configuration of blocks in the dependable system diagram, in the final configuration. In the forth level, the behavior of each block is modeled by means of EDSPN, which generates numeric or analytic expressions of reliability, availability and safety attributes. Finally, the obtained expressions are used in a dependable and parameterized model (MDP), according to the configuration defined in the third level, for obtaining dependability estimates of the system as a whole.

The proposed methodology, not only allows automation, but also aims to hide the mathematical complexity involved and to reduce the state space explosion possibility.

To make the models EDSPN and MDP more efficient, a model library is created. A same model, with the aid of different configuration parameters, can assume different fault tolerant mechanisms, which makes the methodology flexible, as well as, a same model,

with the aid of structural parameters, can assume different levels of redundancy in a same fault tolerant mechanism, which makes the methodology expandable.

Keywords: fault tolerance, dependability, stochastic models, EDSPN.

Sumário

1 Introdução	1
1.1 Contextualização	2
1.2 Objetivo	7
1.3 Justificativas	7
1.4 Alguns Trabalhos Relacionados	9
1.5 Contribuições	11
1.6 Plano de Tese	12
2 Fundamentação	13
Introdução	13
2.1 Definição de Sistemas e Modelos	13
2.2 Conceitos Básicos de Variáveis Aleatórias e Processos Estocásticos	18
2.3 Conceitos de Redes de Petri	30
2.3.1 Propriedades Comportamentais das Redes de Petri	37
2.3.2 Técnica de Análise Qualitativa por Regras de Redução Simples	38
2.3.3 Síntese de redes de Petri	39
2.3.3.1 Métodos de Composição Modular ou Síntese <i>Bottom-up</i>	39
2.3.3.2 Síntese <i>Top-down</i>	41
2.3.3.3 Síntese Híbrida	41
2.4 Dependabilidade e Tolerância a Falhas	41
2.4.1 Principais Conceitos e Taxonomia	41
2.4.2 Meios de Validação de Dependabilidade	43
2.4.3 Meios de Obtenção de Dependabilidade	44
2.4.4 Atributos de Dependabilidade	45
2.5 Técnicas de Redundância de Hardware	46
2.5.1 Técnicas de Redundância Estática ou Passiva	46
2.5.2 Técnicas de Redundância Dinâmica ou Ativa	49
2.6 Técnicas de Redundância de Software	53
2.6.1 Redundância Estática ou Passiva : Programação de N-Versões (NVP)	53
2.6.2 Redundância Dinâmica ou Ativa : Blocos de Recuperação (RB)	55
Considerações Finais	57
3 Trabalhos Relacionados	59
Introdução	59
3.1 Avaliação Determinística e Estocástica	59
3.2 Métodos Baseados em Modelos Numéricos/Analíticos	60
3.2.1 Modelos Combinatoriais (não baseados na geração do espaço de estados)	60
3.2.2 Modelos baseados na geração de espaço de estados	61

3.2.2.1 Modelos não-Markovianos	62
3.2.2.2 Técnicas de <i>Largeness Avoidance</i>	62
3.2.2.3 Técnicas de <i>Largeness Tolerance</i>	63
3.3 Trabalhos Centrados na Visão do Usuário	63
3.4 Trabalhos utilizando formalismo estocástico, em especial, redes de Petri ...	64
3.5 Novas Tendências	71
Considerações Finais	72
4 Metodologia de Análise e Modelagem de Dependabilidade	73
Introdução	73
4.1 Metodologia de Modelagem, Refinamento e Análise de Sistemas Dependáveis	75
4.2 Etapas da Metodologia	76
4.3 Metodologia – Descrição Geral	78
4.4 Níveis Hierárquicos	83
4.4.1 Nível Hierárquico 1: Diagrama de blocos do sistema	84
4.4.2 Nível Hierárquico 2: Diagramas de blocos de dependabilidade estendida (EDBD)	86
4.4.3 Nível Hierárquico 3: Diagrama de blocos intermediários	92
4.4.3.1 Métodos de composição	93
4.4.4 Nível Hierárquico 4: Modelos EDSPN	100
4.4.4.1 Exemplo I: Modelo EDSPN sem Cobertura de Falhas	102
4.4.4.2 Exemplo II: Modelo EDSPN com Cobertura de Falhas	103
4.4.4.3 Exemplo III: Modelo EDSPN composto por hardware e software	106
4.4.5 Nível Hierárquico 5: Diagrama de sistema dependável e parametrizado	110
4.5 Exemplo ilustrativo da aplicação da metodologia	116
Considerações Finais	120
5 Especificação dos Modelos	121
Introdução	121
5.1 Nível Hierárquico 4 - Modelos EDSPN	121
5.1.1 Modelo EDSPN – Bloco Básico sem Replicação	122
5.1.2 Modelo EDSPN – Bloco Básico com Replicação Passiva <i>ColdStandby</i>	123
5.1.3 Modelo EDSPN – Bloco Básico Com Replicação Passiva <i>WarmStandby</i>	125
5.1.4 Modelo EDSPN – Bloco Básico Com Replicação Semi-Ativa <i>HotStandby</i>	127
5.1.5 Modelo EDSPN – Bloco Básico Com Replicação Ativa NMR	128
5.2 Nível Hierárquico 5: Modelos Dependáveis e Parametrizados (MDP)	130
5.2.1 Modelo MDP: Modelo Bloco Básico Markoviano	132
5.2.2 Modelo MDP: Modelo Blocos de Subsistema	134
5.2.3 Modelo MDP: Modelo Bloco <i>Standby</i>	136
5.2.4 Modelo MDP: Modelo Bloco de Decisão Básico	140
5.2.5 Modelo MDP: Modelo Bloco Serial Múltiplo	146

5.2.6	Modelo MDP: Modelo Bloco Ativo	150
	Considerações Finais	152
6	Refinamento dos modelos EDSPN	153
	Introdução	153
6.1	Modelo EDSPN: Bloco Básico sem Replicação - Refinamento dos Eventos de Falha e de Reparo (Distribuições Não-Markovianas)	153
6.2	Modelo EDSPN: Bloco Básico sem Replicação - Refinamento do Estado de Falha	155
6.3	Modelo EDSPN: Bloco Composto com Refinamento das Taxas de disparo das Transições Temporizadas de Reparo	157
6.4	Modelo EDSPN: Bloco sem replicação - refinamento dos componentes de hardware e software	159
6.4.1	Modelo Bloco Básico: Configuração Sw único em Hw único	159
6.4.2	Modelo Bloco Básico: Configuração Sw único, com Cobertura, em Hw único	161
6.4.3	Modelo Bloco Básico: Configuração Sw múltiplo, em Configuração NMR, e Hw único	162
6.4.4	Modelo Bloco Básico: Configuração Sw múltiplo, em NMR com Cobertura, e Hw único	164
6.4.5	Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (<i>ColdStandby</i>), e Hw único	166
6.4.6	Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (<i>ColdStandby</i>) e Cobertura, e Hw único	168
6.4.7	Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (<i>WarmStandby</i>), e Hw único	170
6.4.8	Modelo Bloco: Configuração Sw múltiplo, com Replicação Passiva (<i>WarmStandby</i>) e Cobertura, e Hw único	171
6.4.9	Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Semi-ativa (<i>HotStandby</i>), e Hw único	172
6.4.10	Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Semi-ativa (<i>HotStandby</i>) com Cobertura, e Hw único	173
6.4.11	Modelo Bloco Básico: Configuração Sw múltiplo e Hw múltiplo	175
	Considerações Finais	178
7	Apresentação de Estudos de Caso	179
	Introdução	179
7.1	Estudo de Caso I: Circuito de disparo do motor de um foguete lançador de mísseis	179
7.2	Estudo de Caso II: Sistema de telecomunicações em suporte a um sistema de transmissão de energia elétrica	183
7.3	Estudo de Caso III: Sistema de controle de vôo de aeronaves	195
7.4	Estudo de Caso IV: Utilização de mecanismo tolerante a falhas na implementação de um sistema de ordenação composto por Hw e Sw.....	203
	Considerações Finais	207

8 Conclusão e Trabalhos Futuros	208
8.1 Conclusão	208
8.2 Trabalhos Futuros	213

ANEXOS

A Processos de Nascimento e Morte	224
A.1 Processo Especial de Nascimento e Morte: Rede de Fila M/M/m	224
A.2 Processo Especial de Nascimento e Morte: Reparador Único	225
A.1 Processo Especial de Nascimento e Morte: Independência estocástica	226
B Sintaxe da Ferramenta de Modelagem	228
B.1 Símbolos Utilizados	228
B.2 Definição da Sintaxe	228

Lista de Figuras

<i>Figura 1.1</i>	<i>Árvore de dependabilidade.....</i>	<i>2</i>
<i>Figura 2.1</i>	<i>Representação de Sistema por Modelo</i>	<i>14</i>
<i>Figura 2.2</i>	<i>Diagrama de estados de um processo de nascimento e morte.....</i>	<i>30</i>
<i>Figura 2.3</i>	<i>Regras de redução.....</i>	<i>39</i>
<i>Figura 2.4</i>	<i>Junção de um conjunto de lugares (1-Way Merge).....</i>	<i>40</i>
<i>Figura 2.5</i>	<i>Definição das classes de falhas.....</i>	<i>42</i>
<i>Figura 2.6</i>	<i>Classificação de erros</i>	<i>42</i>
<i>Figura 2.7</i>	<i>Classificação das ocorrências de defeito.....</i>	<i>42</i>
<i>Figura 2.8</i>	<i>Meios para obtenção e validação da dependabilidade de um sistema</i>	<i>43</i>
<i>Figura 2.9</i>	<i>Redundância Modular Tripla (TMR)</i>	<i>47</i>
<i>Figura 2.10</i>	<i>Redundância TMR Triplicada (TTMR)</i>	<i>47</i>
<i>Figura 2.11</i>	<i>Múltiplos estágios TTMR</i>	<i>48</i>
<i>Figura 2.12</i>	<i>Redundância modular de ordem N.....</i>	<i>48</i>
<i>Figura 2.13</i>	<i>Duplicação com Comparação.....</i>	<i>50</i>
<i>Figura 2.14</i>	<i>Módulo Redundante em Espera a Quente.....</i>	<i>51</i>
<i>Figura 2.15</i>	<i>Módulo Reserva em Espera a Frio ou Morna.....</i>	<i>52</i>
<i>Figura 2.16</i>	<i>Técnica híbrida – NMR com Espera</i>	<i>53</i>
<i>Figura 2.17</i>	<i>Programação a N-Versões (NVP).....</i>	<i>54</i>
<i>Figura 2.18</i>	<i>Estrutura do mecanismo de Bloco de Recuperação.....</i>	<i>55</i>
<i>Figura 2.19</i>	<i>Redundância Dinâmica por Bloco de Recuperação</i>	<i>57</i>
<i>Figura 4.1</i>	<i>Evolução da metodologia: dos modelos de blocos aos modelos MDP.....</i>	<i>75</i>
<i>Figura 4.2</i>	<i>Fases de execução da metodologia de modelagem, avaliação e refinamento... </i>	<i>76</i>
<i>Figura 4.3</i>	<i>Metodologia de modelagem, avaliação e refinamento de sistemas dependáveis</i> <i>.....</i>	<i>78</i>
<i>Figura 4.4</i>	<i>Especificação do Sistema em Diagrama de Blocos</i>	<i>85</i>
<i>Figura 4.5</i>	<i>Diagrama de blocos tolerante a falhas</i>	<i>89</i>
<i>Figura 4.6</i>	<i>Diagrama de blocos tolerante a falhas com bloco de decisão.....</i>	<i>90</i>
<i>Figura 4.7</i>	<i>Bloco Intermediário</i>	<i>92</i>
<i>Figura 4.8</i>	<i>Bloco intermediário de decisão.....</i>	<i>92</i>
<i>Figura 4.9</i>	<i>Processo de composição de blocos em série em bloco composto equivalente... </i>	<i>95</i>
<i>Figura 4.10</i>	<i>Processo de composição de blocos intermediários concorrentes.....</i>	<i>96</i>
<i>Figura 4.11</i>	<i>Processo de composição de blocos intermediários com bloco de decisão</i>	<i>97</i>
<i>Figura 4.12</i>	<i>Processo de composição de blocos intermediários com bloco de decisão</i>	<i>99</i>
<i>Figura 4.13</i>	<i>Diagrama de blocos intermediário correspondente ao diagrama EDBD</i>	<i>100</i>
<i>Figura 4.14</i>	<i>Bloco bem formado.....</i>	<i>101</i>
<i>Figura 4.15</i>	<i>Modelo EDSPN sem cobertura de falhas para o bloco básico X</i>	<i>102</i>
<i>Figura 4.16</i>	<i>Modelo EDSPN com cobertura de falhas para o bloco X.</i>	<i>104</i>
<i>Figura 4.17</i>	<i>Modelo EDSPN para blocos compostos por Hw e Sw</i>	<i>107</i>
<i>Figura 4.18</i>	<i>Modelo MDP do bloco básico funcional.....</i>	<i>111</i>
<i>Figura 4.19</i>	<i>Modelo MDP do bloco de subsistema.....</i>	<i>113</i>
<i>Figura 4.20</i>	<i>Modelo MDP do bloco de decisão</i>	<i>115</i>
<i>Figura 4.21</i>	<i>Diagrama de blocos</i>	<i>116</i>

<i>Figura 4.22 Diagrama EDBD do diagrama de blocos do sistema</i>	<i>117</i>
<i>Figura 4.23 Diagrama de blocos intermediários.....</i>	<i>119</i>
<i>Figura 4.24 Especificação do sistema por modelos dependáveis e parametrizados</i>	<i>119</i>
<i>Figura 5.1 Modelo EDSPN para bloco básico sem replicação</i>	<i>123</i>
<i>Figura 5.2 Modelo EDSPN para o bloco com replicação passiva coldstandby.....</i>	<i>124</i>
<i>Figura 5.3 Distribuição do tempo de vida do bloco básico warmstandby com cobertura perfeita no modelo de estágios de Cox.....</i>	<i>126</i>
<i>Figura 5.4 Distribuição do tempo de vida do bloco básico warmstandby com cobertura imperfeita.....</i>	<i>126</i>
<i>Figura 5.5 Modelo EDSPN para o bloco com replicação semi-ativa hotstandby.....</i>	<i>128</i>
<i>Figura 5.6 Modelo EDSPN para o bloco com replicação ativa NMR.....</i>	<i>129</i>
<i>Figura 5.7 Modelo MDP do bloco básico.....</i>	<i>132</i>
<i>Figura 5.8 Modelo MDP para o bloco de subsistema</i>	<i>135</i>
<i>Figura 5.9 Modelo MDP do bloco standby com estimativas analíticas</i>	<i>138</i>
<i>Figura 5.10 Modelo MDP do bloco standby com estimativas numéricas</i>	<i>140</i>
<i>Figura 5.11 Modelo MDP do bloco de decisão</i>	<i>142</i>
<i>Figura 5.12 Modelo MDP dos blocos seriais múltiplos.....</i>	<i>146</i>
<i>Figura 5.13 Modelo MDP de bloco ativo (blocos paralelos concorrentes)</i>	<i>151</i>
<i>Figura 6.1 Modelo EDSPN de bloco básico com refinamento de evento</i>	<i>154</i>
<i>Figura 6.2 Modelo EDSPN de bloco básico sem replicação com refinamento de lugar ..</i>	<i>156</i>
<i>Figura 6.3 Modelo EDSPN para múltiplos blocos com diferentes soluções de reparo.....</i>	<i>158</i>
<i>Figura 6.4 Modelo bloco único formado por hw único e sw único.....</i>	<i>160</i>
<i>Figura 6.5 Modelo de bloco básico de hw e sw únicos com cobertura de falhas.....</i>	<i>162</i>
<i>Figura 6.6 Modelo de bloco básico com Hw único e Sw em configuração NMR.....</i>	<i>163</i>
<i>Figura 6.7 Modelo de bloco básico com Hw único e Sw NMR com cobertura</i>	<i>165</i>
<i>Figura 6.8 Modelo de bloco com Hw único e Sw com replicação cold e warmstandby....</i>	<i>167</i>
<i>Figura 6.9 Bloco de Hw único e Sw com réplicas cold ou warmstandby com cobertura..</i>	<i>169</i>
<i>Figura 6.10 Modelo de bloco com Hw único e Sw com replicação hotstandby</i>	<i>172</i>
<i>Figura 6.11 Modelo de bloco básico com Hw único e réplicas de Sw em hotstandby</i>	<i>174</i>
<i>Figura 6.12 Modelo de Bloco básico com Hw e Sw múltiplos.....</i>	<i>176</i>
<i>Figura 7.1 Diagrama de blocos de confiabilidade do circuito de disparo do motor.....</i>	<i>179</i>
<i>Figura 7.2 Modelo MDP para não habilitação do circuito de disparo.....</i>	<i>181</i>
<i>Figura 7.3 Modelo MDP para não ativação da bateria.</i>	<i>181</i>
<i>Figura 7.4 Modelo MDP do circuito de disparo do motor do foguete.</i>	<i>182</i>
<i>Figura 7.5 Diagrama de blocos hierárquico e modular da infra-estrutura de comunicação</i>	<i>184</i>
<i>Figura 7.6 Diagrama de blocos dos equipamentos do subsistema de telecomunicações..</i>	<i>185</i>
<i>Figura 7.7 Modelo MDP correspondente ao Sistema Irradiante</i>	<i>186</i>
<i>Figura 7.8 Modelo MDP correspondente ao Multiplexador ADM.....</i>	<i>187</i>
<i>Figura 7.9 Modelo MDP correspondente ao Radio Digital SDH 7,5GHz</i>	<i>188</i>
<i>Figura 7.10 Modelo MDP correspondente aos Trechos I ou II.....</i>	<i>189</i>
<i>Figura 7.11 Gráficos relativos ao trecho I (II) do subsistema com cobertura perfeita....</i>	<i>191</i>
<i>Figura 7.12 Gráficos relativos ao trecho I (II) do subsistema com cobertura imperfeita.</i>	<i>191</i>
<i>Figura 7.13 Modelo MDP do subsistema formado pelos trechos I e II.</i>	<i>193</i>
<i>Figura 7.14 Gráficos relativos ao subsistema com cobertura perfeita.....</i>	<i>193</i>
<i>Figura 7.15 Sistema de controle da aeronave.....</i>	<i>196</i>
<i>Figura 7.16 Sistema de controle de vôo.....</i>	<i>197</i>

<i>Figura 7.17 Arquitetura candidata TTMR</i>	198
<i>Figura 7.18 Arquitetura candidata TTMR – Configuração serial</i>	198
<i>Figura 7.19 Modelos MDP das configurações TMR e TMR com Flux-Summing</i>	199
<i>Figura 7.20 Modelo MDP da configuração TTMR do sistema de controle de vôo</i>	200
<i>Figura 7.21 Arquitetura candidata TDTMR</i>	201
<i>Figura 7.22 Arquitetura candidata 5MR</i>	202
<i>Figura 7.23 Microcontrolador 8051</i>	204
<i>Figura 7.24 Sistema de ordenação com mecanismo TMR</i>	204
<i>Figura 7.25 Simulação do sistema de ordenação por meio da técnica TMR</i>	205
<i>Figura 7.26 Diagrama de blocos intermediário</i>	205
<i>Figura 7.27 Componente de Sw único em componente de Hw único</i>	206
<i>Figura 7.28 Modelo MDP para sistema de ordenação com mecanismo TMR</i>	206
<i>Figura A.1 Diagrama de estados da rede de fila M/M/m</i>	224
<i>Figura A.2 Diagrama de estados do modelo machine repairman</i>	226
<i>Figura A.3 Diagrama do modelo com independência estocástica</i>	227

Lista de Tabelas

<i>Tabela 2.1 (Coeficiente de variação) X (Distribuição).....</i>	<i>20</i>
<i>Tabela 2.2 Símbolos das redes de Petri</i>	<i>32</i>
<i>Tabela 4.1 Expressões analíticas e métricas do modelo EDSPN sem cobertura de falhas</i>	<i>103</i>
<i>Tabela 4.2 Expressões analíticas e métricas do modelo EDSPN com cobertura de falhas</i>	<i>105</i>
<i>Tabela 4.3 Expressões de multiplicidade dos arcos do modelo EDSPN do bloco.....</i>	<i>107</i>
<i>Tabela 4.4 Funções de guarda do modelo EDSPN do bloco X.....</i>	<i>108</i>
<i>Tabela 4.5 Disponibilidade do modelo EDSPN do bloco de Hw e Sw</i>	<i>109</i>
<i>Tabela 4.6 Confiabilidade do modelo EDSPN do bloco de Hw e Sw</i>	<i>109</i>
<i>Tabela 4.7 Lugares do modelo MDP correspondente ao bloco funcional básico</i>	<i>111</i>
<i>Tabela 4.8 Transições do modelo MDP correspondente ao bloco funcional básico.....</i>	<i>112</i>
<i>Tabela 4.9 Expressões de multiplicidade de arcos do bloco funcional básico no MDP ...</i>	<i>112</i>
<i>Tabela 4.10 Transições do modelo MDP correspondente ao bloco de subsistema.....</i>	<i>113</i>
<i>Tabela 4.11 Expressões de multiplicidade de arcos do modelo MDP do subsistema</i>	<i>114</i>
<i>Tabela 4.12 Relação número de tokens e pontos de verificação.....</i>	<i>114</i>
<i>Tabela 4.13 Valores numéricos de confiabilidade para um bloco de subsistema</i>	<i>115</i>
<i>Tabela 4.14 Métricas do sistema dependável e parametrizado</i>	<i>118</i>
<i>Tabela 5.1 Expressões analíticas e métricas de dependabilidade</i>	<i>123</i>
<i>Tabela 5.2 Expressões analíticas e métricas de dependabilidade para o bloco básico com replicação passiva coldstandby.....</i>	<i>125</i>
<i>Tabela 5.3 Expressões analíticas e métricas de dependabilidade para o bloco básico com replicação passiva warmstandby.</i>	<i>126</i>
<i>Tabela 5.4 Expressões analíticas e métricas de dependabilidade para modelo hotstandby</i>	<i>127</i>
<i>Tabela 5.5 Expressões analíticas e métricas de dependabilidade para o modelo NMR....</i>	<i>130</i>
<i>Tabela 5.6 Parâmetros de configuração do modelo MDP do bloco básico</i>	<i>133</i>
<i>Tabela 5.7 Composição dos parâmetros de configuração do modelo MDP bloco básico</i>	<i>133</i>
<i>Tabela 5.8 Peso das transições imediatas em conflito no modelo MDP do bloco básico.</i>	<i>133</i>
<i>Tabela 5.9 Estimativas x Métricas do modelo MDP do bloco básico.....</i>	<i>134</i>
<i>Tabela 5.10 Composição dos parâmetros de controle do modelo MDP de subsistema</i>	<i>135</i>
<i>Tabela 5.11 Pesos das transições do modelo MDP de bloco de subsistema</i>	<i>136</i>
<i>Tabela 5.12 Atributos x Métricas do bloco subsistema.....</i>	<i>136</i>
<i>Tabela 5.13 Definição dos parâmetros de controle do modelo de bloco standby</i>	<i>138</i>
<i>Tabela 5. 14 Composição dos parâmetros de controle do modelo MDP de bloco standby</i>	<i>139</i>
<i>Tabela 5.15 Atributos x Métricas bloco standby.....</i>	<i>139</i>
<i>Tabela 5.16 Parâmetros de configuração do modelo MDP de bloco de decisão</i>	<i>141</i>
<i>Tabela 5.17 Multiplicidade dos arcos de entrada de uma transição de falha segura</i>	<i>143</i>
<i>Tabela 5.18 Multiplicidade dos arcos de entrada de uma transição de falha insegura</i>	<i>143</i>
<i>Tabela 5.19 Multiplicidade dos arcos de entrada de uma transição de sucesso</i>	<i>144</i>
<i>Tabela 5.20 Composição dos parâmetros de controle do modelo de bloco de decisão</i>	<i>145</i>
<i>Tabela 5.21 Atributos x Métricas do bloco de decisão</i>	<i>146</i>

<i>Tabela 5.22 Multiplicidade dos arcos do modelo MDP dos blocos serial múltiplos</i>	147
<i>Tabela 5.23 Composição dos parâmetros de configuração do modelo bloco serial múltiplo</i>	147
<i>Tabela 5.24 Pesos das transições no modelo MDP do bloco serial múltiplo</i>	148
<i>Tabela 6.1 Expressões analíticas e métricas de dependabilidade para o bloco básico sem replicação e execução de falha e reparo em l fases.</i>	155
<i>Tabela 6.2 Expressões analíticas e métricas de dependabilidade para o bloco básico sem replicação e com cobertura de falhas.</i>	157
<i>Tabela 6.3 Expressões lógicas condicionais dos tempos de recuperação</i>	158
<i>Tabela 6.4 Métricas de dependabilidade</i>	159
<i>Tabela 6.5 Multiplicidade dos arcos dependentes das marcações</i>	160
<i>Tabela 6.6 Pesos e funções de guarda das transições imediatas</i>	160
<i>Tabela 6.7 Definição das métricas de dependabilidade</i>	160
<i>Tabela 6.8 Multiplicidade dos arcos dependentes das marcações</i>	161
<i>Tabela 6.9 Pesos e funções de guarda das transições imediatas</i>	161
<i>Tabela 6.10 Definição das métricas com cobertura de falhas</i>	162
<i>Tabela 6.11 Multiplicidade dos arcos dependentes das marcações</i>	163
<i>Tabela 6.12 Pesos e funções de guarda das transições imediatas</i>	163
<i>Tabela 6.13 Definição das métricas</i>	164
<i>Tabela 6.14 Multiplicidade dos arcos dependentes das marcações</i>	165
<i>Tabela 6.15 Pesos e funções de guarda das transições imediatas</i>	165
<i>Tabela 6.16 Definição das métricas com cobertura de falhas</i>	166
<i>Tabela 6.17 Multiplicidade dos arcos dependentes das marcações</i>	167
<i>Tabela 6.18 Pesos e funções de guarda das transições imediatas do modelo coldstandby</i>	167
<i>Tabela 6.19 Definição das métricas com cobertura de falhas</i>	168
<i>Tabela 6.20 Multiplicidade dos arcos dependentes das marcações</i>	168
<i>Tabela 6.21 Pesos e funções de guarda das transições imediatas</i>	169
<i>Tabela 6.22 Definição das métricas com cobertura de falhas</i>	169
<i>Tabela 6.23 Multiplicidade dos arcos dependentes das marcações</i>	173
<i>Tabela 6.24 Pesos e funções de guarda das transições imediatas</i>	173
<i>Tabela 6.25 Definição das métricas</i>	173
<i>Tabela 6.26 Multiplicidade dos arcos dependentes das marcações</i>	174
<i>Tabela 6.27 Pesos e funções de guarda das transições imediatas</i>	174
<i>Tabela 6.28 Definição das métricas com cobertura de falhas</i>	175
<i>Tabela 6.29 Lugares do modelo de blocos múltiplos</i>	176
<i>Tabela 6.30 Transições temporizadas do modelo de blocos múltiplos</i>	177
<i>Tabela 6.31 Pesos e funções de guarda das transições imediatas</i>	177
<i>Tabela 6.32 Multiplicidade dos arcos dependentes das marcações</i>	177
<i>Tabela 6.33 Definição das métricas</i>	178
<i>Tabela 7.1 Probabilidades de ocorrência dos modos de falha</i>	180
<i>Tabela 7.2 Definição dos parâmetros dos diversos dispositivos dos trechos I e II</i>	186
<i>Tabela 7.3 Confiabilidade com cobertura perfeita de falhas</i>	190
<i>Tabela 7.4 Disponibilidade com cobertura perfeita de falhas</i>	190
<i>Tabela 7.5 Fatores de cobertura dos componentes</i>	191
<i>Tabela 7.6 Valores de dependabilidade para cobertura de falhas imperfeita</i>	192
<i>Tabela 7.7 Disponibilidade com cobertura perfeita de falhas</i>	192

<i>Tabela 7.8 Confiabilidade do subsistema com cobertura perfeita de falhas</i>	<i>194</i>
<i>Tabela 7.9 Disponibilidade do subsistema.....</i>	<i>194</i>
<i>Tabela 7.10 Estimativas de dependabilidade da configuração TTMR</i>	<i>199</i>
<i>Tabela 7.11 Estimativas de dependabilidade da configuração TDTMR</i>	<i>202</i>
<i>Tabela 7.12 Estimativas de dependabilidade da configuração 5MR.....</i>	<i>203</i>
<i>Tabela 7.13 Confiabilidade dos blocos do sistema de ordenação</i>	<i>207</i>
<i>Tabela 7.14 Comparação de resultados: modelo x simulador.....</i>	<i>207</i>
<i>Tabela A.1 Notação de Kendall.....</i>	<i>224</i>

Lista de Abreviações

BDD	<i>Binary Decision Diagram</i>
CDF	<i>Cumulative Distribution Function</i>
COTS	<i>Commercial-Off-The-Shelf</i>
CTMC	<i>Continuous Time Markov Chain</i>
DCPN	<i>Dynamically Coloured Petri Net</i>
DRB	<i>Distributed Recovery Block</i>
DTMC	<i>Discrete Time Markov Chain</i>
EDBD	<i>Extended Dependability Block Diagram</i>
EDSPN	<i>Extended Deterministic Stochastic Petri Net</i>
EMI	<i>Eletronegnetic Interference</i>
FCFS	<i>First Come First Served</i>
FT	<i>Fault Tree</i>
FTRE	<i>Fault Tree Repeated Events</i>
GreatSPN	<i>Graphical editor and analyzer for timed and stochastic Petri nets.</i>
GSPN	<i>Generalized Stochastic Petri Net</i>
MDP	<i>Modelo de Dependabilidade Parametrizado</i>
METFAC	<i>Ferramenta de análise de confiabilidade e performabilidade desenvolvida pela Universidade Politécnica da Catalunya</i>
MRGP	<i>Markov Regenerative Process</i>
MRM	<i>Markov Reward Model</i>
MRRM	<i>Markov Reward Regenerative Model</i>
MTBF	<i>Mean Time Between Failure</i>
MTEP	<i>Mean Time To Error Perception</i>
MTTF	<i>Mean Time To Failure</i>
MTTR	<i>Mean Time To Repair</i>
NMR	<i>N-Modular Redundancy</i>
NUMAS	<i>NUmerical Methods for the Analysis of computer Systems</i>
NVP	<i>N-Version Programing</i>
PDF	<i>Probability Density Function</i>
PFQN	<i>Product-Form Queueing Networks</i>
PN	<i>Petri Net</i>
QN	<i>Queueing Networks</i>
RB	<i>Recovery Block</i>
RBD	<i>Reliability Block Diagram</i>
RF	<i>Radio Frequency</i>
RG	<i>Reliability Graph</i>
SAN	<i>Stochastic Activity Net</i>
SAVE	<i>System Availability Estimator</i>
SHARPE	<i>Symbolic Hierarchical Automated Reliability and Performance Evaluator</i>
SMP	<i>Semi-Markov Process</i>
SPN	<i>Stochastic Petri Net</i>
SPNP	<i>Stochastic Petri Net Package</i>

SRN	<i>Stochastic Reward Net</i>
SURF2	<i>A program for dependability evaluation of complex hardware and software systems</i>
TANGRAM	Ferramenta de análise de confiabilidade e performabilidade desenvolvida pela Universidade da Califórnia
TimeNet	Ferramenta de modelagem e avaliação de dependabilidade, desenvolvida pela Technische Universität Berlin
TMR	<i>Triple Modular Redundancy</i>
UltraSAN	Ferramenta de análise de confiabilidade e performabilidade desenvolvida pela Universidade do Arizona

Capítulo 1

Introdução

O crescimento do conhecimento técnico e científico tem estimulado o desenvolvimento de novos produtos em nossa sociedade, em especial daqueles produtos que tem no computador o seu elo principal, seja de supervisão, controle, processamento ou comunicação. A contínua e rápida evolução tecnológica tem proporcionado elevados incrementos na capacidade computacional dos processadores, por preços cada vez mais reduzidos. Estes avanços mudaram o foco dos grandes sistemas *mainframes*, constituídos por um único processador de alta capacidade de processamento, para sistemas multiprocessadores distribuídos. Isto tem incrementado sobremaneira a quantidade e a utilização de sistemas computacionais, individualmente ou em redes, o que tem provocado uma série de novas tendências, que impactam na dependabilidade dos sistemas computacionais [127]: a) mudança nas fontes de erro; b) aumento da complexidade; c) grande número de dispositivos computacionais e redes em uso. Os custos anuais com paralisações ou mesmo falhas dos sistemas computacionais são da ordem de bilhões de dólares devido à dependência cada vez maior das pessoas e das organizações nos sistemas de computação, em especial nas redes de computadores [71]. Uma rápida paralisação, por menor que seja, pode implicar em uma infinidade de negócios cancelados e grande insatisfação por parte dos usuários.

A medida que os sistemas de computação e de comunicação permeiam todos os espaços da vida cotidiana das pessoas, as necessidades de se avaliarem a segurança de funcionamento (dependabilidade) de tais sistemas demonstram uma importância crescente. Grande parte desta demanda provém do acentuado nível de exigência dos consumidores, interessados cada vez mais em produtos com preços mais competitivos e de melhor qualidade. A qualidade requerida não deve ser satisfeita apenas pela inclusão de novos serviços, mas pela garantia de funcionamento correto dos serviços existentes. A demanda por melhor qualidade se explica pela responsabilidade que o ser humano deposita nos produtos manufaturados, especificamente nos sistemas computadorizados ou microcomputadorizados [58]. Falhas nestes componentes podem produzir desde um simples inconveniente, como o não funcionamento de um telefone celular, até situações catastróficas como acidentes aéreos ou a explosão de uma usina nuclear. Enquanto razões econômicas forçam o desenvolvimento de novos sistemas computacionais com um número cada vez maior de facilidades, razões de qualidade impõem a necessidade de que sejam evitados maus funcionamentos desses sistemas. Por este motivo, técnicas de avaliação e de modelagem têm provado ser uma solução útil e versátil em todas as fases do ciclo de vida de um sistema dependável [111].

1.1 Contextualização

Desde o surgimento dos primeiros computadores, as aplicações que demandam uma maior confiabilidade, se defrontam com problemas os mais diversos. As dificuldades podem ser encontradas tanto no hardware, em relação a qualidade dos componentes empregados, quanto no software, em relação a qualidade da especificação e do projeto desenvolvidos, ou ainda em relação à troca de informação entre processadores num ambiente de rede sujeito à interferências diversas, especialmente eletromagnéticas. Até meados da década de 80, a maior parte dos sistemas dependáveis eram orientados à missão, demandando altos requisitos de confiabilidade, como os computadores das naves espaciais, dos sistemas de aviação e dos sistemas de defesa antimísseis. A partir de então, passou-se a observar os sistemas dependáveis de operação contínua, com elevados requisitos de disponibilidade, tais como os sistemas de comutação telefônicos, os sistemas computacionais de uso geral e as redes de comunicação e serviços.

Os sistemas analisados nesta Tese são classificados como sistemas dependáveis [80] ou de funcionamento seguro [69]. Segundo [80], sistemas dependáveis são aqueles nos quais confiança possa ser justificadamente depositada nos serviços por eles liberados. De uma forma mais geral, define-se qualitativamente a dependabilidade de um sistema, como sendo a habilidade para liberação de um serviço que possa justificadamente ser confiado [14]. Quantitativamente, segundo [14], pode-se definir a dependabilidade de um sistema como a habilidade para evitar defeitos dos serviços que são mais frequentes e mais severos do que é aceito para os usuários. O grau de confiança a ser depositado nos serviços liberados por um sistema dependável pode ser justificado pela análise de vários critérios científicos integrados, tais como confiabilidade, disponibilidade, segurança, integridade, confidencialidade, manutenibilidade e segurança contra intrusão.

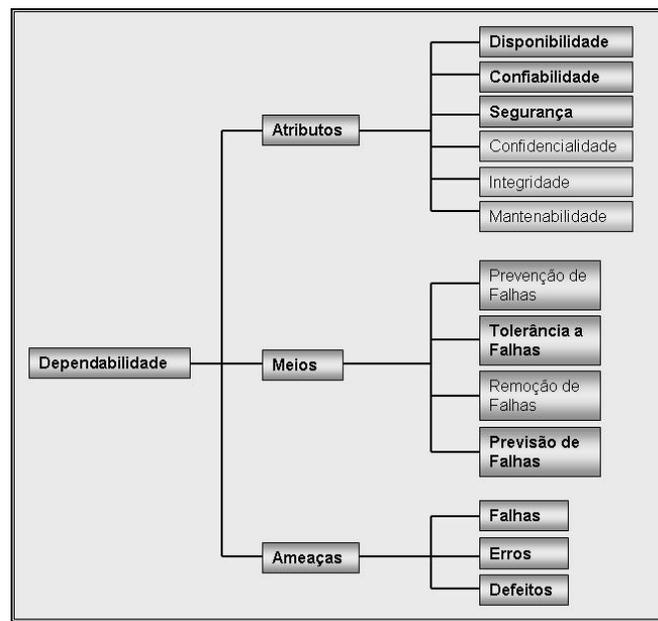


Figura 1.1 Árvore de dependabilidade

Conforme pode ser observado na Figura 1.1, os sistemas dependáveis estão constantemente sujeitos à várias formas de ameaças, representadas por falhas, erros e defeitos. Como forma de garantir os níveis de dependabilidade requeridos, estes sistemas devem utilizar diversos meios de proteção, representados por mecanismos de prevenção, tolerância, remoção e previsão de falhas, cuja efetividade de uso pode ser quantificada através da avaliação dos atributos associados ao sistema.

Dentre os diversos meios utilizados para prover o sistema com o nível de dependabilidade requerido, são de interesse nesta Tese, os mecanismos de tolerância e de previsão de falhas. Podem-se definir os mecanismos de tolerância a falhas como aqueles que procuram preservar a liberação da função do sistema, apesar da presença de falhas ativas, enquanto os mecanismos de previsão de falhas são aqueles que objetivam a realização de avaliação do comportamento do sistema com relação a ocorrência dessas falhas. Dependendo do ambiente da aplicação, um ou mais critérios podem refletir o comportamento do sistema dependável. Dentre os vários atributos que compõem o conceito de dependabilidade, esta Tese está particularmente interessada nos critérios de confiabilidade, disponibilidade e segurança. Os mecanismos de proteção e os atributos de dependabilidade de interesse, destacados em negrito na Figura 1.1, serão detalhados no capítulo de fundamentação teórica.

Dois aspectos são fundamentais nas metodologias de avaliação e modelagem dos sistemas dependáveis: a) o desenvolvimento de modelos que sejam fiéis ao comportamento do sistema e que sejam matematicamente tratáveis; b) o desenvolvimento de procedimentos de avaliação que permitam o processamento do modelo. Modelos estocásticos são em geral requeridos para a representação dos sistemas dependáveis, uma vez que os aspectos relacionados à dependabilidade lidam com as estruturas dos sistemas (número de unidades operacionais de um determinado recurso), e estas devido as inter-relações por vezes complicadas, a ausência de informações detalhadas e a imprevisibilidade de situações no mundo real, tornam a observação de tais processos aleatória [67].

Técnicas de avaliação de dependabilidade de sistemas computacionais podem ser orientadas a medições ou a modelos, de acordo com a classificação seguinte:

- *avaliação determinística*: baseada em medições, executa medições sobre um sistema real, em uso, ou sobre um protótipo do sistema por meio de injeção de falhas. Injeção de falhas permite compreender o efeito das falhas sobre o sistema e avaliar a eficiência dos mecanismos de tolerância a falhas. Injeção de falhas é recomendado para novos sistemas ou para componentes comerciais do tipo COTS (*Commercial-Off-The-Shelf*) para os quais haja pouca ou nenhuma informação de dependabilidade disponível do campo;
- *avaliação estocástica*: baseada em modelos, executa medições em modelos abstratos e simplificados que levam em conta apenas os detalhes relevantes do sistema, de modo a permitir uma maior tratabilidade matemática. Os modelos estocásticos podem ser classificados, de forma abrangente, por mecanismos de simulação ou por métodos

de solução numérico/analíticos, os quais envolvem a solução de expressões analíticas de sistemas de equações lineares, ou sistemas de equações integrais e diferenciais [86] [64]. Os métodos de avaliação estocástica podem ser divididos em:

- *métodos de solução numérica*: atuam sobre todo o espaço de estados do processo estocástico inerente ao modelo do sistema. Fornecem os resultados mais acurados dentre todos os métodos estocásticos, porém sofrem do problema da explosão do espaço de estados. Este tipo de método é denominado método baseado na geração do espaço de estados (*state-space method*) [22];
- *métodos de forma fechada (closed-form) e combinatoriais* [140]: não requer a geração do espaço de estados do processo estocástico e, portanto, não sofre do problema da explosão de estados. Apesar do baixo custo computacional e da eficiência desses métodos, sua aplicação é limitada a um pequeno conjunto de problemas reais, dada as restrições necessárias para a obtenção das expressões analíticas que representam os sistemas e aos aspectos de dependência estocástica e cobertura de falhas;
- *métodos numéricos aproximados*: introduzem mudanças no modelo de modo a simplificar a solução computacional. Como exemplos desses métodos, podem-se citar os métodos de decomposição hierárquica [140] e os métodos de aproximação por fase [88]. A qualidade desses métodos está na definição da aproximação requerida;

As avaliações baseadas em modelos podem dispor de uma série de modelos analíticos, alguns dos quais vêm sendo pesquisados e utilizados durante décadas, e que de uma forma geral, podem ser classificados como modelos combinatoriais e modelos de espaço de estados. As técnicas combinatoriais podem ser representadas por diagramas de blocos de confiabilidade (*Reliability Block Diagram ou RBD*) [136] [70] [139] [126] [141] [101] [17], árvores de falha (*Fault Trees ou FT*) e suas extensões [140] [139] [17] [47] [74][75] [90] [106] [26], grafo de confiabilidade (*Reliability Graphs ou RG*) [140] [139] [89] [101], e redes de filas da forma produto (*product-form queueing networks - PFQN*) [22]. As técnicas de espaço de estados podem ser representadas pelas cadeias de Markov de tempo discreto (DTMC) ou contínuo (CTMC) [22] [140] [18] [43] [31], por meio das redes de Petri e suas extensões [105] [31], redes de Petri estocásticas (SPN) [21] [95] [15], redes de Petri estocásticas generalizadas (GSPN) [91] [94] [90], redes de Petri determinísticas e estocásticas (DSPN) [92] [37] [135] e redes de Petri determinísticas e estocásticas estendidas (EDSPN) [86] que serão utilizadas na elaboração dos modelos básicos de dependabilidade desta Tese, devido a sua flexibilidade. Algumas extensões das redes de Petri também são bastante utilizadas, como as *Stochastic Reward Nets* (SRN) [36] e as *Stochastic Activity Nets* (SAN) [124]. De modo a suportar a representação dos modelos em redes de Petri e suas extensões, muitas ferramentas computacionais têm sido criadas e estão disponíveis, como por exemplo: TimeNet [134] [59] que suporta os modelos GSPN, DSPN e EDSPN; UltraSAN [123] que suporta os

modelos SAN; GreatSPN [33] que suporta GSPN; SPNP [34] que suporta SRN; Surf2 [16][132] que suporta as cadeias de Markov e os modelos GSPN.

Este trabalho tem como objetivo não apenas lidar com a complexidade e a dinamicidade dos sistemas atuais, permitindo que avaliações possam ser feitas em sistemas já existentes, de modo a melhorar os seus critérios de dependabilidade, como também possibilitar a avaliação de arquiteturas candidatas à implementação de sistemas, baseados em modelos que satisfaçam restrições de dependabilidade ainda na fase de projeto, conforme pode ser visto no Capítulo 7, referente aos estudos de caso. Para lidar com a complexidade dos sistemas, são empregados os conceitos de modularização, por meio da divisão do sistema em partes, através de diagramas de blocos, e hierarquia, através da divisão da complexidade do sistema em submodelos independentes, os quais são analisados e os resultados transferidos para os níveis hierárquicos superiores. Para lidar com a dinamicidade dos sistemas é utilizado o conceito de parametrização, onde dependendo do valor das variáveis, como por exemplo, o número de componentes redundantes e o tipo de mecanismo de tolerância a falhas adotado, diferentes arquiteturas podem ser analisadas.

Por meio da introdução de mecanismos de tolerância a falhas nos diagramas de blocos dos sistemas, diferentes arquiteturas podem ser avaliadas. Por meio da parametrização dos modelos estocásticos EDSPN e dos modelos dependáveis e parametrizados, ou simplesmente MDP, e de suas interconexões, diferentes arquiteturas poderão ser modeladas de um modo flexível e rápido. A solução por meio de EDSPN torna o modelo mais conciso, e permite a representação de várias características do sistema de uma forma mais compacta [145].

Nos dias atuais, vários são os problemas que afetam os sistemas. O uso massivo de dispositivos computadorizados fixos e móveis, dentro de um conceito de computação ubíqua, e a crescente pervasividade das redes de computadores e serviços, tornam os sistemas extremamente complexos e dinâmicos. Em muitos casos, tais sistemas devem operar de forma integrada, contínua e de um modo dependável, por exigências legais ou da própria atividade. Associado à complexidade e a dinamicidade dos sistemas, surge um terceiro elemento cada vez mais fundamental: o usuário. A utilização dos dispositivos computadorizados por usuários de diferentes perfis, aumenta a probabilidade que falhas sejam cometidas no sistema de forma involuntária ou malevolente [71]. Portanto, no desenvolvimento de metodologias de avaliação e modelagem, três importantes fatores devem ser levados em consideração: os dispositivos, as redes e os usuários.

A necessidade de uma maior dependabilidade dos sistemas computacionais tem se tornado cada vez mais evidente em virtude de algumas tendências observadas nos últimos anos [126][127]: a) utilização dos sistemas em ambientes cada vez mais severos, sujeitos à variações de temperatura e umidade relativa do ar, poeira, flutuações no fornecimento de energia elétrica e interferências eletromagnéticas; b) massificação dos sistemas digitais, em especial dos sistemas computacionais, e sua utilização por um grande número de usuários sem a devida qualificação, que acarretam em um maior risco de operação inadequada; c) custos de manutenção crescentes; d) sistemas com um número cada vez

maior de componentes, que resulta em maior probabilidade de falha; e) custo proibitivo das paralisações dos sistemas computacionais, devido ao processo de integração com as diversas redes, fixas e móveis, de comunicações e serviços; f) complexidade cada vez maior das interações entre hardware e software, possivelmente com redundância. Um outro problema observado nos dias atuais é a de que previsões feitas para uma década não serão válidas para as décadas seguintes, o que implica que as decisões têm que ser tomadas com rapidez, com bastante antecedência e os sistemas devem ser expansíveis, flexíveis e reusáveis para se adaptar as transformações, sem que tenham que ser totalmente refeitos. Um outro problema que os modelos devem retratar está relacionado ao número de reparadores, a possibilidade dos reparos serem realizados concorrentemente ou sequencialmente, aos tempos médios e aos desvios padrões do tempo de reparo médio.

Neste trabalho assume-se que os modelos são representados por processos estocásticos de estados finitos, em um alto nível de abstração, através do formalismo das redes de Petri estocásticas. Este formalismo procura focar mais no sistema modelado do que nos detalhes de modelagem presentes nos níveis mais baixos, tornando, por conseguinte, os modelos mais compactos e menos sujeitos a erros. Dentre os formalismos das redes de Petri estocásticas, ênfase será dada ao modelo EDSPN, porém sem levar em consideração as transições determinísticas. Deste modo, ter-se-á a habilidade para lidar com os conceitos de modularidade, hierarquia e refinamento de modelos, além de poder utilizar expressões condicionais de multiplicidade de arcos, pesos e retardos dependentes das marcações, na construção de modelos compactos e parametrizados. Num nível de abstração inferior, estas redes de Petri estarão associadas a processos estocásticos, cujo espaço de estados é de natureza discreta, e cujo parâmetro indexador, representado pelo tempo t , é de natureza contínua, nas denominadas Cadeias de Markov de Tempo Contínuo (CTMC). A natureza não-Markoviana de alguns eventos deverá ser representada, nesta Tese, por meio de distribuições exponenciais em fase, ou ainda por sua expressão analítica, desde que suportada pela ferramenta de modelagem [145][134]. Deste modo, além dos problemas de natureza conjuntural, descritos anteriormente, problemas inerentes aos tipos de modelos que são utilizados no desenvolvimento das metodologias de avaliação e modelagem dos sistemas dependáveis, também devem ser considerados. Metodologias que utilizam modelos de espaço de estados, como os modelos CTMC e sua representação por redes de Petri, podem apresentar problemas com relação a explosão de estados, a *stiffness* [101][20], e ao comportamento não-Markoviano associado às distribuições de falha e reparo.

A metodologia a ser utilizada deve permitir ao usuário (modelador) a tomada de decisões relativas a escolha de um dentre vários modelos que satisfaçam as restrições de dependabilidade impostas ao sistema, em especial as restrições relacionadas a confiabilidade, disponibilidade e segurança, considerando-se os mecanismos de tolerância a falhas adotados e a quantidade de componentes redundantes presente em cada um desses mecanismos. Além disso, a metodologia deve ser flexível para que possa lidar com os problemas atuais e com situações futuras. Por fim a metodologia deve ser passível de automação, para utilização por pessoal não-especialista em redes de Petri, pela ocultação da complexidade matemática e dos detalhes de modelagem.

Portanto, é fundamental a avaliação de dependabilidade dos sistemas com relação aos atributos requeridos, por meio de metodologias que permitam a identificação de gargalos naqueles sistemas que se encontram operacionais, ou a escolha de uma entre várias arquiteturas candidatas para aqueles sistemas que se encontram na fase de projeto, de acordo com critérios de confiabilidade, disponibilidade e segurança.

1.2 Objetivo

Este trabalho tem como principal objetivo a definição de uma metodologia que possibilite a modelagem, o refinamento e a avaliação de dependabilidade dos sistemas, de um modo flexível e expansível, buscando ocultar a complexidade matemática envolvida e procurando reduzir a possibilidade de explosão de estados. A metodologia deve permitir a incorporação de técnicas de tolerância a falhas e ser passível de automação.

Alguns dos objetivos específicos a serem obtidos por esta Tese são:

- definição de uma biblioteca de modelos básicos, considerando-se as possibilidades de redundância e refinamento;
- definição de técnica de avaliação de estado transiente por meio de estados permanentes;
- definição de mecanismos de composição de modelos e resultados para possibilitar uma avaliação em níveis de representação, de forma a reduzir a complexidade inerente ao processo de avaliação;
- definição de modelos e regras de reparo de acordo com estratégias de reparo.

1.3 Justificativas

Ao longo dos anos várias técnicas vêm sendo utilizadas para avaliação de dependabilidade. Técnicas de diagrama de blocos de confiabilidade, árvore de falha, grafo de confiabilidade, redes de filas, *model checking*, álgebra de processos e especialmente redes de Petri e suas extensões [106], têm sido bastante empregadas.

As soluções estocásticas se justificam devido à aleatoriedade das ocorrências das falhas e da impossibilidade prática de se analisar deterministicamente um sistema que possua grande quantidade de parâmetros, ou que esteja em estágios iniciais de projeto [86]. Os diagramas de blocos são eficientes tanto para especificação quanto para avaliação de sistemas, porém não lidam de forma eficiente com dependências de reparo e cobertura de falhas[106][17]. Por sua vez, os modelos Markovianos apesar de capturar tais comportamentos, podem gerar espaço de estados de dimensões que inviabilizem sua análise. A aplicação dos modelos Markovianos para análise de dependabilidade é um procedimento tedioso e sujeito a erros, especialmente quando o número de estados torna-

se muito grande. Ademais, os modeladores sentem dificuldade em traduzir seus problemas para as cadeias de Markov [140]. Dentre os formalismos de modelagem desenvolvidos para lidar com os problemas das cadeias de Markov, anteriormente citados, optou-se nesta Tese pelo uso das redes de Petri estocásticas determinísticas estendidas (EDSPN) [86].

A opção do formalismo de modelagem por meio de redes de Petri, nesta Tese, observa as seguintes vantagens:

- disponibilidade de ferramentas gráficas e matemáticas que permitem a descrição formal e análise do comportamento do sistema, levando-se em conta comportamentos concorrentes, síncronos ou assíncronos, distribuídos, não-determinístico e estocástico;
- formalismo com semântica bem estabelecida e métodos de construção e tratamento de modelos complexos bem definidos, que permitem a construção de modelos concisos, e que são úteis no desenvolvimento de metodologias de modelagem e análise de sistemas dependáveis;
- ferramentas que permitem a integração dos conceitos de análise de dependabilidade com a descrição do sistema propriamente dita;
- formalismo que permite não apenas análises das características estruturais, por meio de matriz de incidência e pela estrutura gráfica do modelo, como também análises comportamentais por meio das propriedades de limitação, ausência de deadlock, segurança, *liveness*, reversibilidade e alcançabilidade, em qualquer fase do desenvolvimento do sistema;
- formalismo que apresenta uma grande disponibilidade de métodos de análise matemáticos para solução dos problemas dos modelos Markovianos, os quais são bem definidos e estabelecidos;
- formalismo que incorpora a noção de estado e regras para mudança de estados que permite a captura de características estáticas e dinâmicas de um sistema real;
- formalismo empregado no suporte de muitos aspectos do desenvolvimento de sistemas complexos de diversos campos de atividade [94]: a) especificação formal; b) verificação; c) prototipação rápida; d) documentação;
- formalismo de modelagem bastante maduro na exploração teórica e no desenvolvimento de ferramentas de avaliação de dependabilidade;
- formalismo de grande flexibilidade e de alto poder de abstração, possibilitando a introdução e a adaptação de elementos gráficos e/ou matemáticos com a finalidade de aproximar usuários de sistemas mais específicos.

Por outro lado, as redes de Petri estocásticas por terem suas análises baseadas em modelos de níveis mais baixos, como os modelos CTMC, para a computação das estimativas de dependabilidade, apresentam as seguintes desvantagens [22][89]:

- *largeness*: dimensão do espaço de estados, ou do conjunto de alcançabilidade, algumas vezes proibitivo para análise;
- *stiffness*: presença de atributos de tempo de diferentes ordens no modelo, dificultando possíveis soluções numéricas;
- *atividades não-Markovianas*: dificuldade de modelar atividades ou sistemas que não sigam o modelo Markoviano, com relação a tratabilidade matemática.

1.4 Alguns Trabalhos Relacionados

Modelagem requer o conhecimento da arquitetura do sistema, por intermédio dos seus componentes e das interações entre eles, os mecanismos de detecção de erros e de tolerância a falhas, e as políticas de manutenção. Nos modelos de dependabilidade, os fenômenos associados são representados por suas taxas de ocorrência (falha, reparo, percepção) ou probabilidades condicionais (fator de cobertura). O principal problema apresentado pelas cadeias de Markov, que representam o comportamento dos sistemas complexos nos níveis de abstração mais baixos, é o número de estados. Várias técnicas têm sido desenvolvidas para lidar com este problema, as quais podem ser agrupadas em duas categorias [22]:

- técnicas de *largeness avoidance*: estas técnicas são caracterizadas pela redução do espaço de estados. Como exemplo, pode-se citar, o método de composição hierárquica e agregação [79], o qual particiona os modelos complexos em uma hierarquia de submodelos e substitui um conjunto de estados, num bloco de partição, por um único estado;
- técnicas de *largeness tolerance*: são técnicas que têm como objetivo o domínio da complexidade do modelo por meio de métodos de especificação concisos e de geração automática do modelo. Estes métodos utilizam estruturas de dados especiais, como por exemplo, a estrutura de dados *Binary Decision Diagram* (BDD) [25], e/ou representações para reduzir o espaço de estados.

Especificamente com relação as rede de Petri, a maior parte dos métodos são organizados em duas classes [1]: a) decomposição/agregação; b) construção modular/composição. A primeira classe se caracteriza por uma decomposição hierárquica para evitar a geração de grandes espaços de estados. Nesta classe, o modelo geral é substituído por um conjunto de sub-modelos de modo a facilitar o tratamento matemático. As medidas obtidas pelos vários sub-modelos independentes são então agregadas de modo a calcular as medidas do modelo geral. Como exemplo do método de decomposição hierárquica e agregação pode-se citar o método de composição

hierárquica, organizado em níveis hierárquicos, o qual é aplicado aos sistemas de missões por fases [102]. Do ponto de vista prático, esta técnica é eficiente quando os sub-modelos são fracamente acoplados e torna-se difícil de implementar quando as interações são complexas. A segunda classe é mais adequada na modelagem de sistemas com grande número de componentes fortemente conectados. A idéia básica é a geração do modelo de um sistema de uma forma modular, pela composição dos sub-modelos de seus componentes, por meio de técnicas de composição. Estas técnicas fazem uso de regras de composição para interligação de sub-modelos, de modo a facilitar a geração de modelos, dominar a complexidade e preservar as propriedades formais. Como exemplo, a técnica de supressão das transições imediatas de uma rede GSPN em [5] permite a redução do espaço de estados. A técnica de composição por blocos [73] consiste na construção modular e sistemática do modelo de dependabilidade de um sistema através da composição de sub-modelos dos componentes de hardware, de software e da interações entre eles.

A técnica de composição hierárquica, aplicada aos sistemas de missão por fases [102] é organizado em dois níveis hierárquicos. O nível superior, correspondente a missão, tem cada fase do nível inferior como um evento. O comportamento de cada fase, por sua vez, é detalhado no nível inferior. A técnica de modelagem descendente é proposta em [4]. Nesta técnica a arquitetura do sistema que se pretende modelar não é conhecida no momento da modelagem. As arquiteturas do sistema podem ter diferentes números de componentes, diferentes técnicas de tolerância a falhas e diferentes quantidades de informação. O método de modelagem descendente é composto por dois níveis: nível funcional e nível estrutural. Os modelos começam a ser desenvolvidos a partir do nível funcional e são refinados por etapas. Este método permite a modelagem de diferentes sistemas com poucas alterações do modelo.

A metodologia que está sendo proposta combina aspectos de modularização e hierarquia, do modelo de composição hierárquica, com a flexibilidade e praticidade de construção do modelo de modelagem descendente, com uma vantagem: por ser parametrizado, diferentemente dos demais, aceita componentes com diferentes taxas de falha λ_n , taxas de reparo μ_n e fatores de cobertura C_n . Diferentemente das duas metodologias citadas, as configurações dos sistemas na metodologia proposta são representadas por meio de regras lógicas condicionais dependentes das marcações. Com isto, os modelos tornam-se mais compactos e flexíveis. Com a metodologia proposta, pode-se ter vários modelos representados por um único modelo, porém com valores de parâmetros diferentes. O modelo NMR (*N-Modular Redundancy*), por exemplo, representa todos os modelos de tolerância a falhas com redundância estática n . Para se trocar um modelo com tolerância a falhas NMR por outro, basta alterar o valor de n , o que vêm ao encontro da flexibilidade e praticidade que se espera das metodologias no acompanhamento das evoluções tecnológicas. Assim como na metodologia de modelagem descendente, a metodologia proposta não necessita saber de antemão detalhes dos componentes que irão compor a arquitetura. A avaliação de dependabilidade por parte da metodologia proposta fornece valores de confiabilidade, disponibilidade e segurança, em estado permanente ou transitório, de uma forma rápida, com a geração de um número reduzido de estados, apesar da complexidade dos sistemas a serem modelados. Assim

como na metodologia de composição hierárquica, os valores dos parâmetros a serem utilizados em um nível hierárquico, são obtidos pela solução dos modelos de nível inferior. Na metodologia proposta os valores podem ser expressões numéricas ou expressões analíticas (funções matemáticas), o que permite a utilização de componentes cujas distribuições sejam analiticamente conhecidas, diretamente no modelo geral do sistema, sem a necessidade de geração e análise de um modelo no nível inferior. A metodologia em questão é capaz de modelar blocos formados por hardware e software, em diversas configurações, assim como ocorre na metodologia de modelagem descendente, porém usando modelos distintos para o hardware e para o software, e cujas dependências entre eles são controladas por regras lógicas condicionais dependentes das marcações, o que torna a análise mais concisa e menos complexa. Diferentemente das metodologias citadas, a metodologia proposta faz a avaliação da confiabilidade e da disponibilidade por meio de avaliações de estado permanente [53]. A metodologia objetiva satisfazer aos critérios definidos em [129] para composição de modelos, os quais têm como meta a melhoria das avaliações de dependabilidade.

1.5 Contribuições

Dentre as contribuições do trabalho, estão as seguintes:

- a) desenvolvimento de uma metodologia que possibilita a avaliação, a modelagem e o refinamento de sistemas dependáveis de um modo flexível e passível de automação, ocultando a complexidade matemática envolvida e permitindo a incorporação de mecanismos de tolerância a falhas ao sistema. Esta metodologia pode ser aplicada aos sistemas em operação ou ainda na fase inicial de projeto, com redução espaço de estados e da característica de *stiffness*;
- b) definição de uma biblioteca de modelos desenvolvidos para níveis hierárquicos distintos. Os modelos levam em conta, as formas de replicação passiva, semi-ativa e ativa, utilizadas na modelagem de um grande número de sistemas;
- c) desenvolvimento de modelos únicos para representação de mecanismos tolerante a falhas, com diferentes níveis de redundância e diferentes taxas de falha e reparo, ou ainda, o desenvolvimento de modelos únicos para representação de diferentes mecanismos tolerante a falhas;
- d) possibilidade de avaliação de análise transiente, através de uma seqüência de análises de estado permanente do sistema, considerando-se intervalos de tempo regulares, denominados pontos de verificação;
- e) possibilidade de avaliação, através de métodos de composição modular, de diagramas que não apresentem configurações exclusivamente serial e paralela;
- f) a definição de mecanismos de composição de modelos e resultados que possibilitam uma avaliação em diferentes níveis hierárquicos;

- g) possibilidade de avaliação das estimativas de dependabilidade do diagrama do sistema, por partes, facilitando desta forma a depuração do modelo geral do sistema.

1.6 Descrição da Tese

A fundamentação teórica, a ser apresentada no próximo capítulo descreve os conceitos básicos e a terminologia relativa a sistemas e modelos, variáveis aleatórias, processos estocásticos Markovianos, técnicas de modelagem, redes de Petri, dependabilidade e tolerância a falhas. No Capítulo 3, uma revisão é feita dos trabalhos relacionados à metodologia de modelagem e avaliação de sistemas dependáveis proposta, especialmente com relação ao problema da explosão de estados dos modelos. O Capítulo 4 apresenta a metodologia de modelagem e avaliação dos sistemas dependáveis. Nos Capítulos 5 e 6 são descritos uma série de modelos EDSPN, os quais compõem uma biblioteca de modelos básicos, a serem utilizados no desenvolvimento de novos modelos ou para representação dos blocos que compõem o sistema em análise. Ainda no Capítulo 5, são descritos os modelos MDP básicos, cujas conexões entre si, obedecem as mesmas configurações dos blocos do diagrama de sistemas EDBD (*Extended Dependability Block Diagram*). Os modelos MDP, com o auxílio de regras lógicas e parâmetros de configuração, podem modelar diversas configurações para o sistema e gerar os valores de dependabilidade correspondentes. No Capítulo 6 são definidas algumas possibilidades de refinamento dos modelos, as quais são relativas aos componentes de cada bloco, aos seus estados, eventos e conexões. No Capítulo 7 são apresentados alguns estudos de caso. Os dois primeiros estudos de caso estão relacionados com sistemas já em operação, o terceiro estudo de caso está relacionado com a seleção de uma arquitetura candidata, para implementação de um sistema ainda na fase de projeto e um quarto estudo de caso está relacionado com a implementação do mecanismo de tolerância a falhas, TMR, por meio de componentes de hardware e software. Finalmente, no Capítulo 8, são apresentadas as considerações finais e as pesquisas futuras.

Capítulo 2

Fundamentação

Introdução

O desenvolvimento de uma metodologia de avaliação de sistemas dependáveis, por meio de redes de Petri estocásticas, está fundamentado em uma série de conceitos e definições. Inicialmente, serão abordadas diversas classificações de sistemas e conceitos básicos a respeito da teoria de sistemas. Num mundo cada vez mais dependente dos computadores, observa-se que muitos dos processos utilizados são acionados por eventos instantâneos, os quais podem ocorrer continuamente, ou em intervalos de tempos, muitas vezes de um modo não previsível. Isto nos conduz à definição das variáveis aleatórias contínuas e discretas, cujas famílias quando indexadas por um parâmetro tal como o tempo, definem o que se chama de processos estocásticos. No universo dos processos estocásticos ênfase maior será dada aos processos que dependem apenas do momento presente, os chamados processos Markovianos, representados pelas distribuições exponenciais. Uma classe especial dos processos Markovianos, denominada processos de nascimento e morte será rapidamente abordada, os quais, por possuírem uma estrutura especial, facilitam a definição de expressões associadas ao atributo de disponibilidade. Apesar das distribuições exponenciais serem matematicamente tratáveis, as análises das cadeias de Markov podem ser complexas e trabalhosas. De modo a facilitar o processo de modelagem e análise, torna-se necessária a utilização de ferramentas de modelagem de mais alto nível de abstração. As redes de Petri de natureza estocástica, por suas características, satisfazem estes requisitos. A utilização das redes de Petri estocásticas, por meio de modelos EDSPN, para modelagem dos eventos de falha e de reparo dos sistemas dependáveis, faz uso de mecanismos de tolerância a falhas como forma de satisfazer os requisitos de dependabilidade dos sistemas. De modo a permitir uma melhor compreensão dos capítulos seguintes, faz-se necessário a definição de alguns conceitos básicos, além da definição da taxonomia, relativos à dependabilidade e aos mecanismos de tolerância a falhas.

2.1 Definição de Sistemas e Modelos

A definição de sistema é um conceito primitivo que pode dar margens a diversas explicações. Sistema pode ser definido como um conjunto de componentes que atuam juntos para realizar uma determinada função, a qual não seria possível realizar com quaisquer das partes individuais [31], ou como um conjunto de componentes os quais interagem sob o controle de um projeto [6], ou como um conjunto de entradas e saídas, uma funcionalidade, e possivelmente um conjunto de componentes que implementam esta funcionalidade, a qual consiste em um ou mais algoritmos [57] ou ainda é uma entidade que interage com outras entidades, i.e., outros sistemas, incluindo hardwares, softwares, seres humanos e o mundo físico com seus fenômenos naturais [14].

Independentemente da definição, observa-se que elas estão relacionadas a componentes que interagem e à função que se pretende realizar. De um modo mais objetivo pode-se dizer que sistema é algo real como, por exemplo, um automóvel ou um conjunto de dispositivos eletrônicos para controle de vôo das aeronaves. Sistemas reais podem apresentar uma complexidade de tal ordem, em virtude da excessiva quantidade de variáveis a eles associadas, que se torna inexecutável a sua análise. Para que se possa analisá-los sugere-se, nestes casos, a utilização de modelos, os quais são de natureza mais abstrata. São exemplos de modelos as equações matemáticas que descrevem um determinado comportamento do sistema quantitativamente, ou a maquete de um prédio a ser construído. Para executar o processo de modelagem, ou seja, o processo de transformação do sistema em modelo, conforme apresentado na Figura 2.1, deve-se definir um conjunto de variáveis mensuráveis associadas ao dado sistema. Quanto maior o número de variáveis, mais refinado (mais próximo da realidade) se tornará o modelo, porém à custa de uma complexidade que poderá torná-lo inviável para análise [31][49].

Um modelo é composto por um conjunto de variáveis de entrada e de saída e por relações matemáticas que as envolvem, conforme pode ser observado na Figura 2.1.

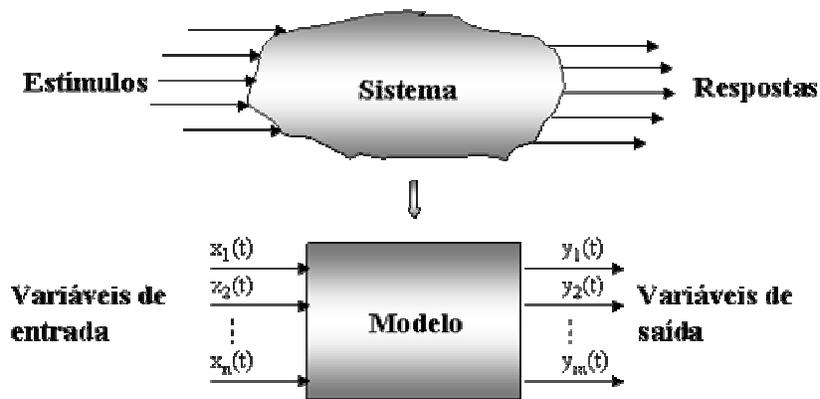


Figura 2.1 Representação de Sistema por Modelo

Uma série de conceitos bem fundamentados a respeito de sistemas e modelos podem ser obtidos em [31].

De acordo com a Figura 2.1, podem-se definir formalmente as n variáveis de entrada através do vetor coluna $\mathbf{u}(t)$, as m variáveis de saída por meio do vetor coluna $\mathbf{y}(t)$, e a relação matemática entre elas, representada por $\mathbf{y}=\mathbf{f}(\mathbf{u})$, por intermédio das relações matemáticas seguintes:

$$\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_n(t)]^T \quad (2.1)$$

$$\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_m(t)]^T \quad (2.2)$$

$$\mathbf{y} = \mathbf{f}(\mathbf{u}) = [f_1(u_1(t), \dots, u_n(t)), \dots, f_m(u_1(t), \dots, u_n(t))]^T \quad (2.3)$$

A semântica das variáveis de entrada e de saída baseia-se em pontos de vista particulares e também nas restrições impostas pela aplicação. As restrições objetivam a redução do número de variáveis do modelo, de modo a facilitar a sua implementação. Quanto mais próxima da realidade, maior será o número de variáveis e conseqüentemente

a complexidade do modelo. Levando-se em conta a dependência da função de saída em relação a função de entrada, os sistemas poderão ser classificados como:

Def. 2.1 Sistemas estáticos (ou sem memória): um sistema é dito estático, ou sem memória, quando a sua saída $y(t)$ independe dos valores de entrada que ocorreram antes do tempo t , isto é, independe de $u(\tau)$, para $\tau < t$;

Def. 2.2 Sistemas dinâmicos (com memória): um sistema é dito dinâmico quando a sua saída $y(t)$ depende dos valores de entrada ocorridos anteriormente ao tempo t , ou seja, é um sistema que possui memória.

Levando-se em consideração a dependência da saída $y = f(u)$ sobre a variável tempo, os sistemas podem ser classificados como:

Def. 2.3 Sistemas estacionários ou invariantes no tempo: um sistema é dito estacionário, quando a saída $y(t)$, resultante da aplicação da entrada $u(t)$, é idêntica a saída $y(t-\tau)$ resultante da aplicação da entrada $u(t-\tau)$, independentemente do valor do deslocamento de tempo τ .

Def. 2.4 Sistemas variantes no tempo: um sistema é dito variante no tempo quando depende não apenas das variáveis de entrada como também da variável tempo, ou seja, $y = f(u, t)$

Além das variáveis de entrada e de saída, representadas por $u(t)$ e $y(t)$, respectivamente, deve-se definir uma nova variável denominada variável de estado, representada pelo vetor $x(t)$. O estado de um sistema está relacionado ao comportamento deste, em um modo mensurável, num determinado instante de tempo. O estado de um sistema no tempo t_0 é a informação requerida em t_0 tal que a saída $y(t)$ seja determinada de um modo único por meio desta informação e da entrada $u(t)$, para todo $t \geq t_0$. Os componentes do vetor de estado $x_1(t), x_2(t), \dots, x_k(t)$, definem as variáveis de estado. Um processo de modelagem por meio de equações matemáticas, envolvendo o vetor de entrada $u(t)$, o vetor de saída $y(t)$ e o vetor de estado $x(t)$, definem a dinâmica de um sistema, cujo modelo é especificado por:

- **Equações de estado:** especifica o estado $x(t)$, dada um vetor de entrada $u(t)$ e um estado inicial $x(t_0)$, $\forall t \geq t_0$;
- **Espaço de estados:** especifica o conjunto de todos os possíveis valores de estado assumidos pelo sistema.

O espaço de estado é completamente especificado pelas seguintes equações:

$$\bullet \text{ Equação de estado: } dx(t)/dt = g(x(t), u(t), t), \text{ e } x(t_0) \text{ o estado inicial} \quad (2.4)$$

$$\bullet \text{ Equação de saída: } y(t) = f(x(t), u(t), t) \quad (2.5)$$

Onde as funções g e f são denominadas função de estado e função de saída, respectivamente. Quando não houver variação de estado do sistema com o tempo, i.e., quando o sistema for estático, tem-se $dx(t)/dt = 0 \forall t$, e o modelo é descrito apenas pela equação de saída. Para sistemas invariantes no tempo, tem-se:

- Equação de estado invariante no tempo: $dx(t)/dt = g(x(t), u(t)) = 0$ (2.6)

- Equação de saída invariante no tempo: $y(t) = f(x(t), u(t))$ (2.7)

Os sistemas, com relação à natureza das funções g e f usadas no modelo, podem ser classificados como linear e não-linear. Um sistema é dito linear quando as funções de estado e de saída, g e f , respectivamente, são lineares. Considerando-se que os sistemas sejam invariantes no tempo e que haja n variáveis de entrada, m variáveis de saída e k variáveis de estado, pode-se definir \mathbf{A} como uma matriz $k \times k$, \mathbf{B} como uma matriz $k \times n$, \mathbf{C} como uma matriz $m \times k$ e \mathbf{D} como uma matriz $m \times n$, cujas entradas definem os parâmetros do modelo, conforme mostrado a seguir:

- Equação de estado: $dx(t)/dt = \mathbf{A}x(t) + \mathbf{B}u(t)$ (2.8)

- Equação de saída: $y(t) = \mathbf{C}x(t) + \mathbf{D}u(t)$ (2.9)

A aplicação da transformada de Laplace às equações de estado e de saída permite a obtenção de equações algébricas ao invés das equações diferenciais e soluções no domínio da frequência, definido pela variável s , ao invés das soluções no domínio do tempo. As transformadas de Laplace [140] são bastante utilizadas em uma série de definições relacionadas a sistemas dependáveis. Um outro conceito importante na análise de sistemas dinâmicos é o conceito de *sample paths*, o qual define a evolução dos estados do sistema com o tempo, para uma determinada entrada. Pode-se verificar a acessibilidade de um determinado estado ou conjunto de estados, a partir de um estado inicial, conforme poderá ser verificado nos grafos de alcançabilidade dos modelos de redes de Petri estocástica.

Levando-se em consideração a natureza do espaço de estados, os modelos podem ser classificados como:

Def. 2.5 Espaço de estado contínuo: consiste em todos os vetores k -dimensionais de números reais, onde k representa o número de variáveis de estado;

Def. 2.6 Espaço de estado discreto: consiste em um conjunto de estados discretos;

Muitas vezes não se pode determinar com exatidão o valor da variável de entrada $u(t)$, $\forall t \geq t_0$, em virtude da ocorrência de uma série de fatores que fogem ao controle, como por exemplo, a ocorrência de descargas atmosféricas ou de outros fatores naturais que possam interferir no valor de entrada proposto. Tais condições de falha na entrada $u(t)$ fazem com que o estado e a saída do sistema se tornem imprevisíveis. Levando-se em conta a previsibilidade das variáveis de saída, os sistemas podem ser classificados como:

Def. 2.7 Sistemas determinísticos: são aqueles cujas variáveis de saída são previsíveis. O estado $x(t)$ poderá ser avaliado por meio das entradas dadas, para todo $t \geq t_0$;

Def. 2.8 Sistemas estocásticos: são aqueles nos quais pelo menos uma das variáveis de saída do sistema não é previsível. O estado no tempo t é uma variável aleatória avaliada por meio de funções de distribuição probabilísticas;

Os sistemas quanto ao processamento e ao transporte da informação podem ser divididos em duas categorias [67]:

Def. 2.9 Sistemas de tempo-real: são aqueles que utilizam modelos de tempo determinístico por causa das ações que ocorrem dentro de intervalos de tempo fixo ou em instantes de tempo distintos. Exemplo desses modelos são as redes de Petri temporizadas.

Def. 2.10 Sistemas de compartilhamento de recursos: são aqueles que utilizam modelos de tempo estocásticos, como as redes de Petri estocásticas, devido à disputa (concorrência) por recursos ou por estratégias de serviço, por exemplo.

Considerando-se a possibilidade de ajustes na entrada de controle por meio de informações obtidas do comportamento do sistema, os sistemas podem ser classificados como [31][24]:

Def. 2.11 Sistemas de Malha Aberta: são aqueles que mantêm as entradas fixas, independentemente dos efeitos que elas possam ter nas saídas observadas;

Def. 2.12 Sistemas de Malha Fechada: são aqueles cujas entradas dependem dos efeitos que elas causam na saída. Este processo de realimentação auxilia na seleção da entrada correta para a realização de uma determinada função.

O tempo é uma variável contínua freqüentemente utilizada para representar as condições das equações de estado e de saída nos sistemas do mundo real. Contudo, de modo a prover flexibilidade, velocidade e baixo custo, e também propiciar a adaptação do mundo real à realidade dos computadores digitais, a discretização da variável tempo é essencial. O processo de discretização do tempo consiste na divisão da variável tempo, contínua, em intervalos inteiros não negativos, constantes, e de comprimento IT. O intervalo de tempo IT é denominado intervalo de verificação. Nos modelos de tempo discretos a variável tempo é constituída por uma seqüência de pontos $t_0 < t_1 < t_2 < t_3 < t_4 \dots < t_i < \dots$. Os intervalos de tempo entre dois pontos seqüenciais quaisquer são os mesmos e iguais a IT, i.e., $t_{i+1} - t_i = IT$, para $i = 0, 1, 2, \dots$.

A Equação de estado 2.4 e a Equação de saída 2.5, considerando-se um índice i , correspondente a amostragem da variável de tempo t , são definidas como:

- Equação de Estado: $\mathbf{x}(i + 1) = \mathbf{g}(\mathbf{x}(i), \mathbf{u}(i), i)$, $\mathbf{x}(0) = \mathbf{x}_0$ (2.10)

- Equação de Saída: $\mathbf{y}(i) = \mathbf{f}(\mathbf{x}(i), \mathbf{u}(i), i)$ (2.11)

As equações lineares de tempo discreto aplicadas aos sistemas invariantes no tempo são representadas por:

- Equação de Estado: $\mathbf{x}(i + 1) = \mathbf{Ax}(i) + \mathbf{Bu}(i)$ (2.12)

- Equações de Saída: $\mathbf{y}(i) = \mathbf{Cx}(i) + \mathbf{Du}(i)$ (2.13)

As transições entre estados são causadas pela ocorrência de eventos. Eventos são ações ou ocorrências que ativam estas transições. As maneiras pelas quais os eventos são acionados permitem caracterizar os modelos de sistemas como:

Def. 2.13 Acionamento por tempo: as transições de estado são sincronizadas pelo disparo do *clock* e pela seleção do evento, ou ausência deste, ocorrido no momento do disparo. Estas transições são ditas síncronas.

Def. 2.14 Acionamento por evento: as transições de estado ocorrem assincronamente com o *clock*, isto é, a mudança de estado não está sincronizada com o disparo do *clock*, como no caso anterior. A noção de acionamento por evento está relacionada à noção de interrupção em sistemas computacionais. Estas transições são ditas assíncronas.

Os sistemas orientados a eventos podem ser modelados em tempo discreto ou contínuo. Um dos formalismos mais utilizados para modelagem de eventos discretos é obtido por meio de redes de Petri, a qual poderá ser não temporizada, temporizada e estocástica. Devido a razões técnicas e econômicas, uma grande parte dos sistemas compartilha recursos. O compartilhamento dos recursos conduz a situações de conflito, as quais provocam variações no tempo de serviço e situações de erro, aumentando a complexidade do sistema e tornando a sua representação por meio de modelos determinísticos impossível. Além disso, apesar da falta de informações mais detalhadas do sistema, da complexidade das interconexões e da não previsibilidade de comportamentos individuais, muitas regularidades estatísticas podem ser observadas e modeladas por meio de processos estocásticos.

No estudo dos sistemas dependáveis, o fornecimento dos serviços providos pelo sistema é submetido a todo tipo de falhas, sejam elas físicas ou algorítmicas, causadas pela natureza ou pelo homem. A ocorrência de falhas são eventos de natureza aleatória, isto é, cujo instante não se é possível prever com exatidão. Conforme a Definição 2.8, estes sistemas são de natureza estocástica, e devem ser avaliados de acordo com as regras impostas aos processos estocásticos.

2.2 Conceitos Básicos de Variáveis Aleatórias e Processos Estocásticos

Alguns conceitos básicos de variáveis aleatórias serão inicialmente abordados, e em seguida serão apresentados os conceitos relativos aos processos estocásticos, em especial aqueles relacionados aos processos Markovianos [18][22][140].

Def. 2.15 Variável aleatória: define-se variável aleatória como sendo uma função que mapeia todos os possíveis resultados do experimento, i.e., o espaço amostral do experimento S , no conjunto de números reais \mathfrak{R} , conforme a expressão $X: S \rightarrow \mathfrak{R}$.

Os modelos de dependabilidade apresentados nos próximos capítulos envolvem espaço de estados representados por variáveis aleatórias discretas, e transformações de um estado para outro, em um dado instante de tempo, representadas por variáveis aleatórias contínuas.

Def. 2.16 Variável aleatória discreta: a variável aleatória discreta X é uma função que atribui valores numéricos aos resultados de um experimento. Os valores assumidos por esta função definem um subconjunto finito ou infinito contável dos números reais. O

conjunto de probabilidades associado a esta variável aleatória caracteriza sua função de probabilidade de massa (pmf) cuja notação é dada por $p_X(x)$. Considerando-se uma variável aleatória discreta X , e os valores de probabilidade a ela associados, tem-se:

$$p_X(x) = P(X = x) = \sum_{X(s)=x} P(s) \quad (2.14)$$

onde $s \in S$ são os pontos amostrais, ou seja, os pontos pertencentes ao espaço de resultados ou espaço amostral S do experimento, e $P(s)$ é a probabilidade dos pontos amostrais, ou eventos, cuja variável aleatória X a eles aplicado é igual a x . Para serem válidas, as pmf's devem satisfazer as seguintes propriedades:

$$a) \quad 0 \leq p_X(x) \leq 1, \quad \forall x \in \mathfrak{R}; \quad (2.15)$$

$$b) \quad \sum_i p_X(x_i) = 1 \quad (2.16)$$

Def. 2.17 Variável aleatória contínua: uma variável aleatória contínua X é uma função que pode assumir todos os valores no intervalo $[a, b]$, onde $-\infty < a < b < \infty$. O conjunto de probabilidades associadas a esta variável aleatória caracteriza sua função de distribuição $F_X(x)$, denominada função de distribuição acumulada (CDF), a qual é definida por:

$$F_X(x) = P(X \leq x), \quad -\infty < x < \infty \quad (2.17)$$

Um outro modo de representação do conjunto de probabilidades associadas às variáveis aleatórias contínuas, pode ser dado por meio da função densidade de probabilidade (pdf) $f_X(x)$, definida por :

$$f_X(x) = \frac{dF_X(x)}{dx} \quad \therefore \quad F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt, \quad -\infty < x < \infty. \quad (2.18)$$

As funções de densidade de probabilidade de uma variável aleatória contínua X devem satisfazer as seguintes propriedades:

$$f_X(x) \geq 0 \quad \forall x. \quad (2.19)$$

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad (2.20)$$

Parâmetros usuais das variáveis aleatórias X discretas (contínuas) associados a uma pmf (pdf), e que são úteis em processos de transformação de algumas distribuições são:

- o valor médio, o valor esperado, ou o tempo médio de vida de uma variável aleatória, é dado por:

(variável aleatória discreta):

$$\bar{X} = E[X] = \sum_{x_i} (x_i) \cdot p_X(x_i) = \sum_{x_i} x_i \cdot P(X = x_i) \quad (2.21)$$

(variável aleatória contínua):

$$\bar{X} = E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx \quad (2.22)$$

- o segundo momento central ou variância da variável aleatória X (discreta ou contínua):

$$\sigma_X^2 = Var(X) = \overline{(X - \bar{X})^2} = \overline{(X^2 - \bar{X}^2)}, \quad \text{onde } \sigma_X \text{ é o desvio padrão, e } \bar{X} \text{ o valor médio} \quad (2.23)$$

- o coeficiente de variação (desvio padrão normalizado):

$$C_X = \frac{\sigma_X}{X} \quad (2.24)$$

O coeficiente de variação é um parâmetro essencial nos métodos de representação de funções não-Markovianas por fases (estágios), a serem mostradas mais adiante. A Tabela 2.1 exprime as aproximações dos modelos não-Markovianos, por diferentes tipos de distribuição, em função do coeficiente de variação.

Coeficiente de Variação	Distribuição
$C_X = 1$	Exponencial
$C_X < 1$	Hipoexponencial
$C_X > 1$	Hiperexponencial

Alguns parâmetros utilizados nas análises de dependabilidade e que têm o intuito de reduzir as equações de tempo contínuo, são definidos a seguir, considerando-se um sistema representado por um diagrama de blocos:

- MTTF (*Mean Time to Failure*)** é o tempo esperado para ocorrência da primeira falha do bloco em análise, considerando-se que no tempo zero (referência inicial de tempo), o bloco estava em perfeitas condições. Devido à dificuldade de modelagem analítica, este parâmetro é, em geral, medido ou estimado. Analiticamente, o MTTF é definido como:

$$MTTF = \int_0^{\infty} R(t) dt \quad (2.25)$$

Considerando-se que a distribuição correspondente ao bloco é exponencial, tem-se:

$$MTTF = \int_0^{\infty} R(t) dt = \int_0^{\infty} \exp^{-\lambda t} dt = \frac{1}{\lambda}, \text{ onde } \lambda \text{ é a taxa de falha do bloco.} \quad (2.26)$$

Logo, uma relação direta é estabelecida entre o MTTF e a taxa de falha λ ;

- MTTR (*Mean Time to Repair*)** é o tempo esperado para o reparo de um bloco em falha. Normalmente este parâmetro, assim como o MTTF, é medido ou estimado. Considerando-se que $G(t)$, a função de mantabilidade, represente a probabilidade que o bloco, uma vez em falha, seja reparado no intervalo de tempo de $[0, t]$, e que a função de reparo apresente uma distribuição exponencial, tem-se:

$$MTTR = \int_0^{\infty} 1 - G(t) dt, \text{ onde } G(t) = 1 - \exp^{-\mu t} \text{ e } \mu \text{ é a taxa de reparo do bloco} \quad (2.27)$$

$$MTTR = \int_0^{\infty} 1 - G(t) dt = \int_0^{\infty} \exp^{-\mu t} dt = \frac{1}{\mu} \quad (2.28)$$

Portanto, uma relação direta é estabelecida entre o MTTR e a taxa de recuperação do bloco μ ;

MTBF (Mean Time Between Failure) é o tempo médio entre falhas em um bloco com reparo, sendo derivado da combinação dos processos de falha e reparo. A expressão aproximada do MTBF em função dos parâmetros MTTF e MTTR é dada por:

$$MTBF = MTTF + MTTR \quad (2.29)$$

- **MTEP (Mean Time to Error Perception)** é o tempo médio de recuperação entre falhas não detectadas. As falhas são não detectadas por ausência de mecanismo de detecção ou por falha desse mecanismo. Esse tempo é a soma do tempo de percepção de uma falha não detectada, em geral bastante longa, mais o tempo de reparo propriamente dito.
- **Fator de Cobertura (Coverage Factor)** é a probabilidade que um bloco seja recuperado dado que uma falha tenha sido detectada. É definida ainda como a probabilidade que uma classe particular de falhas seja detectada antes que o bloco, formado por vários componentes, seja totalmente corrompido. O uso do fator de cobertura determina as falhas passíveis de detecção e aquelas que não são detectadas.

Def. 2.18 Função de probabilidade conjunta: é a função associada a um experimento aleatório, a qual envolve diversas variáveis aleatórias conjuntamente. O resultado do experimento aleatório está relacionado aos valores dessas várias variáveis, as quais podem afetar-se reciprocamente. A função de distribuição conjunta correspondente a variável aleatória $X=(X_1, X_2, \dots, X_n)$, é dada por:

(variáveis aleatórias discretas):

$$P (X_1= x_1, \dots, X_n= x_n) \text{ representa a probabilidade que } X_1= x_1, \dots, X_n= x_n. \quad (2.30)$$

(variáveis aleatórias contínuas):

$$F_X(x) = P (X_1 \leq x_1, \dots, X_n \leq x_n) \text{ representa a probabilidade que } X_1 \leq x_1, \dots, X_n \leq x_n. \quad (2.31)$$

Def. 2.19 Variáveis aleatórias Independentes: As variáveis aleatórias X_1, X_2, \dots, X_n são ditas estatisticamente independentes se:

(variáveis aleatórias discretas):

$$P (X_1= x_1, X_2= x_2, \dots, X_n= x_n) = P(X_1= x_1) \cdot P(X_2= x_2) \dots P(X_n= x_n) \quad (2.32)$$

(variáveis aleatórias contínuas):

$$P (X_1 \leq x_1, X_2 \leq x_2, \dots, X_n \leq x_n) = P(X_1 \leq x_1) \cdot P(X_2 \leq x_2) \dots P (X_n \leq x_n) \quad (2.33)$$

Def. 2.20 Probabilidade condicional: expressa a dependência entre variáveis aleatórias por meio de uma condição. As probabilidades condicionais são utilizadas na definição do teorema das probabilidades totais e nas regras de Bayes, os quais por sua vez são fundamentais na avaliação de confiabilidade das estruturas de sistemas não perfeitamente serial/paralelo [136].

(variáveis aleatórias discretas):

$$P (X_1= x_1| X_2= x_2, \dots X_n= x_n) = P(X_1= x_1, X_2= x_2, \dots X_n= x_n) / P(X_2= x_2, \dots X_n= x_n) \quad (2.34)$$

(variáveis aleatórias contínuas):

$$P (X_1 \leq x_1| X_2 \leq x_2, \dots X_n \leq x_n) = P(X_1 \leq x_1, \dots X_n \leq x_n) / P(X_2 \leq x_2, \dots X_n \leq x_n) \quad (2.35)$$

Várias são as relações envolvendo múltiplas variáveis aleatórias. Por exemplo, quando se analisam os momentos de primeira e segunda ordem de distribuições estatísticas do tipo hipoexponencial, Erlang ou hiperexponencial, representadas por distribuições exponenciais em fase, utilizam-se estas relações. Algumas das relações envolvendo múltiplas variáveis aleatórias são:

- Se c_1, c_2, \dots, c_n são constantes arbitrárias, X_1, X_2, \dots, X_n são variáveis aleatórias (não necessariamente independentes), e E o valor médio esperado das variáveis aleatórias, tem-se:

$$E \left[\sum_{i=1}^n c_i X_i \right] = \sum_{i=1}^n c_i E[X_i]. \quad (2.36)$$

- Se as variáveis aleatórias X_1, X_2, \dots, X_n são independentes, logo:

$$E \left[\prod_{i=1}^n X_i \right] = \prod_{i=1}^n E[X_i]. \quad (2.37)$$

Um outro conceito bastante utilizado no processo de modelagem de sistemas dependáveis é o conceito de estatística de ordem.

Def. 2.19 Estatística de Ordem: sejam X_1, X_2, \dots, X_n variáveis aleatórias contínuas mutuamente independentes, identicamente distribuídas, com função de distribuição F e função densidade f . Seja Y_1, Y_2, \dots, Y_n variáveis aleatórias obtidas pela permutação do conjunto X_1, X_2, \dots, X_n em ordem crescente, isto é, $Y_1 = \min\{ X_1, X_2, \dots, X_n \}$ e $Y_n = \max\{ X_1, X_2, \dots, X_n \}$. A variável aleatória Y_k é denominada a k -ésima estatística de ordem. Devido ao fato de X_1, X_2, \dots, X_n serem variáveis aleatórias contínuas, segue-se que $Y_1 < Y_2 < \dots < Y_n$ com probabilidade 1.

As estatísticas de ordem [140][136] exemplificam os tempos de vida de um conjunto de variáveis aleatórias (que podem representar blocos em um diagrama de blocos) assumindo-se diversas configurações. O caso geral é representado pela configuração m/n (m out of n), a ser definida posteriormente, sendo as conexões em série e em paralelo, casos específicos. A obtenção de estimativas de confiabilidade pode ser formalizada considerando-se que o sistema seja constituído por n blocos mutuamente independentes. Considerando-se X_i como sendo o tempo de vida, ou a disponibilidade, do i -ésimo bloco em um sistema composto por n blocos e considerando-se que exatamente j desses blocos venham a falhar no intervalo de tempo $(-\infty, y]$ ou que estejam inoperantes no tempo y , e que $(n-j)$ blocos permaneçam confiáveis ou disponíveis, pode-se definir a

função de distribuição F da variável aleatória Y_k , para os sistemas não reparáveis, como sendo:

$$F_{Y_k}(y) = P(Y_k \leq y) = P(\text{"pelo menos } k \text{ blocos falhem no intervalo } (-\infty, y]) .:$$

$$F_{Y_k}(y) = \sum_{j=k}^n \binom{n}{j} F^j(y) [1 - F(y)]^{n-j}, \quad -\infty < y < \infty, \quad (2.38)$$

no caso dos sistemas reparáveis, trocando-se a denominação da função de indisponibilidade de $F_{Y_k}(y)$ por $U_{Y_k}(y)$, para possibilitar a distinção das expressões analíticas da indisponibilidade daquela referente à inconfiabilidade, tem-se:

$$U_{Y_k}(y) = P(U_k \leq y) = P(\text{"pelo menos } k \text{ blocos não estejam operacionais em } y\text{"})$$

$$U_{Y_k}(y) = \sum_{j=k}^n \binom{n}{j} U^j(y) [1 - U(y)]^{n-j}, \quad -\infty < y < \infty, \quad (2.39)$$

- **Confiabilidade ou Disponibilidade de um diagrama de blocos em uma configuração Série:**

A confiabilidade ou disponibilidade do sistema em série corresponde a confiabilidade ou a disponibilidade do bloco, de um diagrama de blocos, com menor tempo de vida, ou seja, ao tempo de vida da estatística de primeira ordem. Logo:

$Y_k = Y_1$, onde $Y_1 = \min\{X_1, X_2, \dots, X_n\}$, conseqüentemente,

$$F_{Y_1}(y) = \sum_{j=1}^n \binom{n}{j} F^j(y) [1 - F(y)]^{n-j} = 1 - [1 - F(y)]^n, \quad -\infty < y < \infty \quad (2.40)$$

Considerando-se F_{Y_1} a função de distribuição de falha, a confiabilidade $R(t)$ dos blocos em série é dada por:

$$R_{Série}(t) = R_{Y_1}(t) = 1 - F_{Y_1}(t) = 1 - (1 - [1 - F(y)]^n) = [1 - F(y)]^n = R(t)^n \quad (2.41)$$

Considerando-se U_{Y_1} a função de indisponibilidade dos blocos em série, tem-se que a disponibilidade dos blocos em série é dada por:

$$A_{Série}(t) = A_{Y_1}(t) = 1 - U_{Y_1}(t) = 1 - (1 - [1 - U(y)]^n) = [1 - U(y)]^n = A(t)^n \quad (2.42)$$

Caso os tempos de vida ou os tempos em que os componentes permanecem operacionais sejam distintos, tem-se que:

$$R_{Série}(t) = \prod_{i=1}^n R_i(t) \quad (\text{confiabilidade}) \quad \text{e} \quad (2.43)$$

$$A_{Série}(t) = \prod_{i=1}^n A_i(t) \quad (\text{disponibilidade}) \quad (2.44)$$

Considerando-se a função de distribuição de falha dos blocos como sendo uma função exponencial negativa, com taxas de falhas constantes λ_i , tem-se que a confiabilidade do sistema é dada por:

$$R_i(t) = \exp^{-\lambda_i t} \text{ e } R_{\text{Serie}}(t) = \prod_{i=1}^n \exp^{-\lambda_i t} = \exp^{-\lambda t}, \text{ onde } \lambda = (\lambda_1 + \lambda_2 + \dots + \lambda_n) \quad (2.45)$$

Considerando-se as funções de distribuição de falha e reparo de cada bloco como sendo exponencial negativa, e independentes, com taxas de falhas λ_i e de reparo μ_i constantes, tem-se que a disponibilidade instantânea do sistema é dada por:

$$A_{\text{Serie}}(t) = \prod_{i=1}^n A_i(t) = \prod_{i=1}^n \left[\left(\frac{\mu_i}{\lambda_i + \mu_i} \right) + \frac{\lambda_i}{\lambda_i + \mu_i} \exp^{-(\lambda + \mu)t} \right], \quad (2.46)$$

No limite, quando $t \rightarrow \infty$, tem-se:

$$A_{\text{Serie}}(t) = \prod_{i=1}^n \left(\frac{\mu_i}{\mu_i + \lambda_i} \right), \text{ que representa a disponibilidade de estado permanente.} \quad (2.47)$$

- **Confiabilidade ou Disponibilidade do diagrama de blocos em uma configuração Paralela:**

A confiabilidade do sistema em paralelo corresponde à uma confiabilidade maior que a confiabilidade do bloco com maior tempo de vida, enquanto que a disponibilidade do sistema em paralelo corresponde a uma disponibilidade maior que a disponibilidade do bloco com maior disponibilidade. Portanto, a confiabilidade ou disponibilidade do sistema corresponde ao tempo de vida ou disponibilidade da n-ésima estatística de ordem. Logo:

$Y_k = Y_n$, onde $Y_n = \max\{X_1, X_2, \dots, X_n\}$, conseqüentemente,

$$F_{Y_n}(y) = \sum_{j=n}^n \binom{n}{j} F^n(y) [1 - F(y)]^{n-n} = [F(y)]^n [1 - F(y)]^0 = [F(y)]^n, \quad -\infty < y < \infty \quad (2.48)$$

Portanto, a confiabilidade dos blocos em paralelo é dada por:

$$R_{\text{Paralelo}}(t) = R_{Y_n}(t) = 1 - F_{Y_n}(t) = 1 - [F(y)]^n = 1 - [1 - R(t)]^n. \quad (2.49)$$

Caso os tempos de vida dos blocos sejam distintos, tem-se:

$$R_{\text{Paralelo}}(t) = 1 - \prod_{i=1}^n [1 - R_i(t)] \quad (2.50)$$

A disponibilidade dos blocos em paralelo, por sua vez, no instante de tempo t , é dada por:

$$A_{Paralelo}(t) = A_{Yn}(t) = 1 - F_{Yn}(t) = 1 - [F(y)]^n = 1 - [1 - A(t)]^n \quad (2.51)$$

Caso as disponibilidades sejam distintas, tem-se:

$$A_{Paralelo}(t) = 1 - \prod_{i=1}^n [1 - A_i(t)] \quad (2.52)$$

- **Confiabilidade e Disponibilidade do diagrama de blocos em uma configuração m/n :**

A inconfiabilidade de um sistema com configuração m/n , corresponde à probabilidade de que pelo menos m blocos, de um total de n blocos, venham a falhar no intervalo de tempo $(-\infty, y)$ ou que um máximo de $(n-m)$ blocos estejam funcionando convenientemente. Logo:

$$R_{m/n}(t) = 1 - F_{m/n}(t) = 1 - \left[\sum_{j=m}^n \binom{n}{j} F^j(y) [1 - F(y)]^{n-j} \right] \quad (2.53)$$

A indisponibilidade de um sistema com configuração m/n corresponde a probabilidade de que pelo menos m blocos, de um total de n blocos, estejam indisponíveis no tempo t , ou que um máximo de $(n-m)$ blocos estejam operacionais. Logo:

$$A_{m/n}(t) = 1 - R_{m/n}(t) = 1 - \left[\sum_{j=m}^n \binom{n}{j} F^j(y) [1 - F(y)]^{n-j} \right] \quad (2.54)$$

Caso o valor de n seja ímpar obtém-se o caso particular da configuração NMR. Por exemplo, supondo-se que um sistema composto por 3 blocos, organizados numa configuração TMR venha a falhar, são necessários que pelo menos 2 dos 3 blocos venham a falhar. Portanto considerando-se a mesma taxa de falha tem-se:

$$R_{TMR}(t) = 1 - F_{2/3}(t) = 1 - \left\{ \left[\binom{3}{2} F^2(t) [1 - F(t)]^{(3-2)} \right] + \left[\binom{3}{3} F^3(t) [1 - F(t)]^{(3-3)} \right] \right\} \therefore$$

$$R_{TMR}(t) = 1 - \left\{ 3F^2(t)(1 - F(t)) + F^3(t) \right\} = 1 - \left[3(1 - R(t))^2 R(t) + (1 - R(t))^3 \right] = 3R(t)^2 - 2R(t)^3 \quad (2.55)$$

O mesmo pode ser aplicado as estimativas de disponibilidade. Para que o sistema permaneça operacional são necessários que 2 dos 3 blocos concorrentes estejam operacionais. Logo,

$$A_{TMR}(t) = 1 - \left\{ 3U^2(t)(1 - U(t)) + U^3(t) \right\} = 1 - \left[3(1 - A(t))^2 A(t) + (1 - A(t))^3 \right] = 3A(t)^2 - 2A(t)^3 \quad (2.56)$$

Além das configurações seriais, paralelas e m/n , pode-se utilizar a estatística de ordem para se obter expressões de confiabilidade dos sistemas redundantes *cold*, *warm* e

hot standby. Quando o sistema não segue configurações estritamente seriais e/ou paralelas, duas técnicas podem ser utilizadas: teorema das probabilidades totais [140] e aproximações por limites inferior e superior [126].

A confiabilidade de um sistema não plenamente serial/paralelo pode ser desenvolvida através da aplicação do teorema das probabilidades totais ou regra de eliminação [140][136], a um ou mais blocos capazes de particionar o espaço amostral S em dois subconjuntos disjuntos. Para que o sistema esteja operacional é necessário que exista ao menos um caminho em funcionamento entre os pontos terminais do diagrama. Assumindo-se $R_i(t)$ como a confiabilidade do bloco capaz de segmentar o espaço amostral em dois subconjuntos, $R_S(t)$ como a confiabilidade do sistema, X_S como a variável aleatória que indica que o sistema está operante e X_i como a variável aleatória que indica que o bloco, selecionado para expansão, também está operante, tem-se:

$$P(X_S) = P(X_i)P(X_S | X_i) + P(\overline{X_i})P(X_S | \overline{X_i}) \therefore$$

$$R_S(t) = R_i(t)P(X_S | X_i) + (1 - R_i(t))P(X_S | \overline{X_i})$$

Se um determinado bloco i é selecionado para expansão, então dois diagramas serão criados, os quais serão idênticos ao diagrama original, exceto pelo bloco i . Em um dos diagramas o bloco i é colocado em curto-circuito, indicando que o bloco i está operacional, e no outro diagrama o bloco i é colocado em circuito aberto indicando que o bloco i está em falha e, portanto indisponível. A expressão de confiabilidade de ambos os diagramas equivale à expressão analítica composta dos dois termos da equação anterior. O primeiro termo corresponde ao diagrama de blocos considerando o bloco i em curto-circuito; o segundo termo corresponde ao diagrama de blocos considerando o bloco i em circuito aberto. Caso haja algum outro bloco que impossibilite a conversão de um dos dois diagramas em um modelo serial ou paralelo, deve-se selecionar este outro bloco para expansão e usar a mesma estratégia anterior [140]. Caso o diagrama a ser analisado seja tão complexo que impossibilite a avaliação exata da confiabilidade do sistema, aproximações por limite superior e inferior podem ser usadas [126]. Um dos limites superiores consiste em colocar todos os caminhos entre os pontos terminais em paralelo, mesmo tendo-se blocos repetidos em mais de um caminho. Resultados exatos podem ser obtidos através desta estratégia. Limites inferiores podem ser obtidos em função do conjunto mínimo de corte, o qual é uma lista de componentes, tal que a remoção de qualquer componente da lista fará com que o sistema mude de operacional para falho. Maiores detalhes desta técnica podem ser obtidos em [126][48].

Além das definições relativas às variáveis aleatórias discretas e contínuas, algumas definições importantes acerca de uma classe particular de processos aleatórios, denominada *Processos Markovianos*, serão mostradas. Os processos Markovianos são bastante utilizados no desenvolvimento das métricas de dependabilidade. Dentre os processos Markovianos, importância especial é dada às *Cadeias de Markov*, as quais são processos aleatórios de estados discretos e tempo contínuo que possuem a propriedade de ausência de memória. Por meio dessa propriedade afirma-se que o futuro é incondicionalmente independente dos estados passados, uma vez que tenha sido definido

o estado presente. Ou seja, o estado futuro de uma variável aleatória com ausência de memória só depende do estado presente.

Def. 2.20 Processos Estocásticos: processo estocástico é uma família de variáveis aleatórias $\{X(t) | t \in T\}$, definido sobre um dado espaço de probabilidade e indexado pelo parâmetro t , o qual varia sobre um conjunto T , freqüentemente definido como tempo.

Def. 2.21 Processo Markoviano: um processo estocástico $\{X(t) | t \in T\}$, onde a propriedade de ausência de memória é aplicada. Em outras palavras, a distribuição condicional $X(t)$, depende somente do valor imediatamente anterior $X(t_n)$, e não dos demais valores anteriores $X(t_0), X(t_1), \dots, X(t_{n-1})$, para todo $0 = t_0 < t_1 < \dots < t_n < t_{n-1}$:

$$P[X(t) \leq x | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0] = P[X(t) \leq x | X(t_n) = x_n]. \quad (2.57)$$

Quando o processo de Markov aplica-se a um espaço de estados discreto, tem-se as cadeias de Markov. Quando a função de distribuição condicional da equação anterior tem propriedade de invariância com relação à origem do tempo t_n , conforme Def. 2.3, tem-se as cadeias de Markov homogêneas:

$$P[X(t) \leq x | X(t_n) = x_n] = P[X(t-t_n) \leq x | X(t_n - t_n) = x_n] = P[X(t-t_n) \leq x | X(0) = x_n] \quad (2.58)$$

As cadeias de Markov quanto à natureza contínua ou discreta do parâmetro T , em geral tempo, podem ser classificadas como:

- Cadeias de Markov de Parâmetro Discreto, DTMC, são aquelas restritas a um espaço de estado S discreto, finito ou contável infinito, e um espaço de parâmetro discreto T , onde $T \subseteq \mathbb{N}$;
- Cadeias de Markov de Parâmetro Contínuo, CTMC, são aquelas cujo espaço de estado é o mesmo das cadeias de Markov DTMC, porém o espaço de parâmetro T é contínuo.

A função de distribuição exponencial por ser a única que possui a propriedade de ausência de memória entre todas as variáveis aleatórias contínuas, e por ser matematicamente mais tratável, será utilizada como distribuição básica na modelagem de dependabilidade a ser considerada.

A CDF (*cumulative distribution function*) de uma variável aleatória contínua exponencialmente distribuída com taxa de falha λ e suas respectivas propriedades é mostrada a seguir, por ser utilizada com freqüência no processo de modelagem dos sistemas dependáveis.

$$F_X(x) = \begin{cases} 1 - \exp^{-\lambda x}, & 0 \leq x < \infty, \\ 0, & \text{caso contrário} \end{cases} \quad (2.59)$$

$$\text{pdf: } f_X(x) = \lambda e^{-\lambda x}, \quad (2.60)$$

$$\text{Média: } \bar{X} = \frac{1}{\lambda}, \quad (2.61)$$

$$\text{Variância: } \text{Var}(X) = \frac{1}{\lambda^2}, \quad (2.62)$$

$$\text{Coeficiente de variação: } C_X=1 \quad (2.63)$$

Para cálculo das probabilidades de estado das cadeias de Markov, os seguintes algoritmos podem ser considerados:

- algoritmos para cálculo das probabilidades de estado permanente [22];
- algoritmos para cálculo das probabilidades de estado dependentes do tempo ou transientes [22].

Considerando-se a condição de estacionaridade, as cadeias de Markov discretas e contínuas podem ser representadas pelas seguintes equações lineares:

- DTMC: $\mathbf{v} = \mathbf{vP} \therefore \mathbf{0} = \mathbf{v(P-I)}$ (2.64)

onde \mathbf{v} é um vetor das probabilidades de estado estacionário e \mathbf{P} é a matriz de probabilidades de transição de estados e \mathbf{I} é a matriz identidade;

- CTMC: $\mathbf{0} = \boldsymbol{\pi}\mathbf{Q}$, (2.65)

onde $\mathbf{0}$ é um vetor com todas as probabilidades iguais a zero, $\boldsymbol{\pi}$ é o vetor de probabilidade de estado permanente e \mathbf{Q} é a matriz de transição infinitesimal, ou a matriz geradora infinitesimal, cujos elementos são representados por taxas ao invés de probabilidades como no caso DTMC.

Considerando-se a condição de estacionaridade das cadeias de Markov DTMC e CTMC um sistema linear cuja forma geral é dada por $\mathbf{0} = \mathbf{xA}$, onde \mathbf{A} pode ser $(\mathbf{P-I})$ para as DTMC, ou \mathbf{Q} para as CTMC, podem-se solucionar as cadeias de Markov de estado permanente por intermédio de três diferentes soluções:

- **método numérico direto:** são métodos que fornecem resultados exatos. Para sistemas com alguns milhares de estados, os métodos diretos são mais corretos e confiáveis que os demais. Dentre os métodos diretos podem-se citar dois dos mais importantes: método de *eliminação gaussiana* [22] e método de *Grassmann* [76][22]. O método de eliminação gaussiana utiliza um processo de substituição e eliminação para a obtenção de um sistema modificado de equações, equivalentes ao sistema de equações lineares original. A utilização de valores de probabilidades muito baixos, em virtude do elevado número de estados, pode afetar os resultados de forma adversa nos erros de arredondamento e de cancelamentos. O método de Grassmann, uma variante do procedimento anterior, é menos sensível aos erros de arredondamento e cancelamento por evitar subtrações de números aproximadamente iguais. Erros de arredondamento podem ocorrer durante a utilização do método de *Grassmann*, além de propagarem-se e acumularem-se, o que limita a aplicabilidade desse método aos modelos de cadeias de Markov com um número médio de estados;

- **método numérico iterativo:** os métodos iterativos são mais eficientes em questão de espaço e tempo, porém não fornecem nenhuma garantia de convergência. Além disso, a determinação dos limites de erro para o término das iterações nem sempre é tarefa fácil. Para modelos com um grande número de estados, o método iterativo, por ser mais eficiente, é mais utilizado do que os métodos numéricos diretos. Os métodos iterativos por não alterarem a matriz de parâmetros A , não acumulam erros de arredondamento. A principal desvantagem desse tipo de método é a não garantia de sua convergência, e dependendo do método, a taxa de convergência é bastante sensível aos valores de entrada da matriz de parâmetros. Pode-se acelerar a convergência através de uma adequada estimativa inicial, e terminá-la quando os valores obtidos forem próximos do valor exato, de acordo com um nível de tolerância especificado. Vários são os métodos iterativos, dentre os quais podem ser citados o método de potência, o método de Jacobi, o método de Gauss-Seidel e o método SOR (*Successive Over-Relaxation*) descritos em [22]. Por meio da transformação implícita do espaço de estados, soluções aproximadas podem ser obtidas por meio de métodos de agregação e desagregação, representados pelos métodos de Courtois [40][41][22] e pelo método iterativo de Takahashi [133][22].
- **técnicas que fornecem resultados em *closed-form*:** os métodos de *closed-form* [22] são utilizados quando o modelo possui alguma estrutura na formação da matriz, como por exemplo, os processos de nascimento e morte [140]. Processos de nascimento e morte são bastante úteis na modelagem de várias situações na teoria da confiabilidade. Um determinado processo é caracterizado como sendo de nascimento e morte se a cadeia de Markov de tempo contínuo e homogênea que o representa, apresentar taxas constantes λ_i ($i=0, 1, 2, 3, \dots, n$) e μ_i ($i=1, 2, \dots, n$) que satisfaçam as seguintes taxas de transição:

$$\begin{aligned}
 q_{i,i+1} &= \lambda_i, \\
 q_{i,i-1} &= \mu_i, \\
 q_i &= \lambda_i + \mu_i \\
 q_{ij} &= 0 \quad \text{para } |i - j| > 1
 \end{aligned}
 \tag{2.66}$$

As taxas de nascimento e morte no estado i , definidas por λ_i e μ_i , respectivamente, dependem apenas do estado i e não do tempo. Neste tipo de processo, nascimento e morte podem ocorrer simultaneamente e estão representadas no diagrama de estados abaixo. Considerando-se um comportamento de estado permanente na solução do sistema associado ao processo de nascimento e morte, reduz-se bastante a complexidade das equações pela adoção das equações de balanceamento. No estudo da modelagem de sistemas dependáveis modulares considera-se que o sistema está em um estado permanente, quando a taxa dos blocos que falham é igual a taxa dos blocos que são recuperados em cada estado. No diagrama da Figura 2.2 admite-se uma CTMC de estados finito, n .

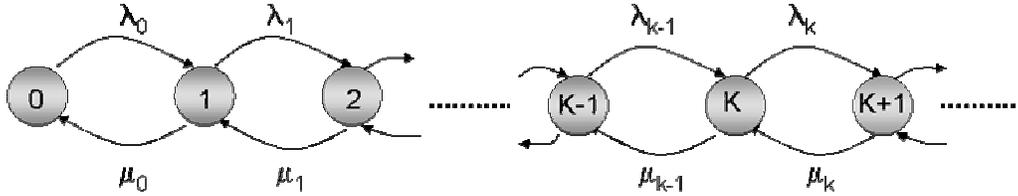


Figura 2.2 Diagrama de estados de um processo de nascimento e morte

Assim sendo, as equações do sistema reduzem-se às equações de balanceamento abaixo, onde π_k , λ_k e μ_k são a probabilidade, a taxa de falha e a taxa de reparo para o estado k , respectivamente:

$$\begin{aligned}
 0 &= -(\lambda_k + \mu_k)\pi_k + \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1} \quad \therefore \\
 (\lambda_k + \mu_k)\pi_k &= \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1} \quad 1 \leq k \leq n-1 \\
 0 &= -\lambda_0\pi_0 + \mu_1\pi_1 \quad \therefore \lambda_0\pi_0 = \mu_1\pi_1 \quad \therefore \lambda_0\pi_0 - \mu_1\pi_1 = 0 \quad \therefore \pi_1 = \left(\frac{\lambda_0}{\mu_1}\right)\pi_0 \quad (2.67)
 \end{aligned}$$

Resolvendo-se a equação anterior para um número k de nós, onde $1 \leq k \leq n$, tem-se:

$$\pi_k = \left(\frac{\lambda_{k-1}\lambda_{k-2}\dots\lambda_0}{\mu_k\mu_{k-1}\dots\mu_1}\right)\pi_0 = \pi_0 \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}}\right), \quad k = 1, 2, \dots, n \quad (2.68)$$

$$\text{Onde, } \pi_0 = \frac{1}{1 + \sum_{k=1}^n \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}}\right)} \quad (2.69)$$

Mais informações a respeito dos processos de nascimento e morte, e sua correlação com as políticas de reparo, podem ser encontrados no anexo-A. Informações a respeito dos processos de nascimento e morte e suas variantes podem ser encontrados em [140].

2.3 Conceitos de Redes de Petri

A utilização das CTMC para análise de confiabilidade e disponibilidade impõe uma série de passos: abstrair o sistema físico, construir a cadeia de Markov, definir as equações diferenciais ordinárias para soluções transitórias ou equações lineares para soluções de estado permanente manualmente, e escrever um programa para a solução numérica das equações [140]. A síntese manual da matriz geradora infinitesimal para sistemas longos e complexos é, em geral, uma tarefa exaustiva. A utilização de métodos automáticos, para especificação e geração da CTMC associada ao sistema, torna-se essencial. As redes de Petri estocásticas, surgidas no final da década de 80, são bastante úteis neste aspecto [139]. Elas são mais concisas em sua especificação e mais intuitivas. Vários pacotes de software foram desenvolvidos para geração automática e solução dos

sistemas estocásticos Markovianos, tais como, TimeNet[134], utilizada nesta Tese, DSPNexpress[86], SHARPE[122], dentre outros.

As redes de Petri foram originalmente introduzidas por Carl Petri em seu destacado trabalho *Kommunikation mit Automaten* [108]. As redes de Petri, de um modo geral, são ferramentas gráficas para descrição formal de sistemas, cujas dinâmicas são caracterizadas por concorrência, sincronização, exclusão mútua e conflito. Estas redes incorporam a noção de estado e regras para mudança de estados, além do conceito do princípio da localidade, o que as permite capturar características estáticas e dinâmicas de um sistema real [94].

Formalmente definem-se as redes de Petri como grafos direcionados constituídos por dois vértices: lugares e transições. Lugares são utilizados para descrever uma condição Booleana ou uma situação. Transições são usadas para descrever eventos ou atividades capazes de alterar os estados do sistema. As transições podem ser imediatas, sem tempo associado, ou temporizadas, com atribuição de tempo exponencial. As transições temporizadas, por exemplo, podem definir tempos médios para falha e reparo associados aos blocos que compõem um sistema, enquanto as transições imediatas, podem descrever a probabilidade de detecção de falhas desses blocos. A inclusão da variável tempo no conceito das transições possibilita a avaliação quantitativa dos requisitos de dependabilidade por meio das redes de Petri [86].

As relações entre lugares e transições são especificadas por arcos. Arcos de entrada conectam lugares a transições e indicam os lugares em que o evento pode ocorrer. Arcos de saída conectam transições a lugares e indicam as modificações nos lugares induzidas pelos eventos. Existem ainda os arcos inibidores que conectam lugares a transições e que indicam lugares capazes de impedir a ocorrência do evento. Pode-se associar uma multiplicidade a estes arcos, a qual, nesta Tese, é representada por um pequeno traço transversal ao arco e um número ou símbolo associado ao mesmo. Quando não existe número ou símbolo associado ao arco a sua multiplicidade é 1.

Os elementos até então descritos compõem a estrutura estática das redes de Petri. A presença do elemento *token* permite as redes de Petri um comportamento dinâmico por meio da especificação de estados. Definem-se marcações ou estados como os conjuntos das possíveis atribuições de *tokens* aos lugares da rede. Partindo-se de uma condição inicial, *tokens* podem ser criados ou destruídos pela execução dos eventos. O número de *tokens* em um determinado lugar P é descrito pela notação #P, conforme o ANEXO-B. A representação simbólica dos elementos de uma rede de Petri é mostrada na Tabela 2.2.

Uma transição está habilitada na marcação corrente se os lugares de entrada desta transição tem uma quantidade de *tokens* no mínimo igual as multiplicidades dos arcos de entrada correspondentes. O disparo de uma transição é uma operação atômica, na qual *tokens* são removidos dos lugares de entrada e depositados nos lugares de saída, de acordo com as multiplicidade dos arcos de entrada e saída, respectivamente, associados a esta transição [100].

Os lugares definem uma condição Booleana quando a quantidade de *tokens* neles presentes é sempre zero ou um. A condição é verdadeira quando há presença de *token*, e falsa, caso contrário. Os lugares também podem definir uma situação, quando as quantidades de tokens neles presentes descrevem esta situação [94]. Situações podem ser especificadas por *tokens*, quando se pretende selecionar uma dentre várias opções de multiplicidade de arcos, de retardos das transições temporizadas, de pesos em transições imediatas em conflito, ou de qualquer outra forma de dependência da marcação. Nos modelos desenvolvidos, os lugares tanto assumem condições Booleanas, definidas como *flags*, quanto situações condicionais dependentes das marcações.

Tabela 2.2 Símbolos das redes de Petri

Símbolo	Descrição
	Lugar
	Transição imediata
	Transição temporizada com distribuição exponencial
	Arco de entrada ou arco de saída
	Arco inibidor
	Arco com multiplicidade 2
	Arco com multiplicidade dependente da expressão em m_1

Dependendo dos tipos de transições suportadas pelo formalismo de modelagem, diferentes possibilidades de redes Petri são possíveis. Quando as transições presentes na rede de Petri são transições temporizadas, e estas estão associadas a retardos exponencialmente distribuídos têm-se as redes de Petri estocásticas (SPN)[15][94]. Quando as transições presentes no grafo são dos tipos imediatas e exponencialmente distribuídas têm-se as redes de Petri estocásticas generalizadas (GSPN) [94]. As redes de Petri determinísticas e estocásticas DSPN [86], formam uma classe particular das redes de Petri estocásticas, as quais incluem transições temporizadas exponencialmente distribuídas e determinísticas, além das transições imediatas. Quando as redes de Petri incluem em seu formalismo definições de arcos, pesos e retardos dependentes das marcações, de modo a representar as características do sistema de um modo mais compacto, tem-se as redes de Petri EDSPN[134][145]. As redes SRN [37], uma extensão das redes GSPN, associam a cada transição tangível uma taxa de recompensa (*reward rate*) as quais permitem que uma série de medidas possam ser especificadas e calculadas usando um formalismo conveniente. Várias extensões utilizadas nas SRN também são encontradas no formalismo de modelagem EDSPN através da ferramenta TimeNet. O modelo EDSPN pode ser formalmente representado pelos elementos do modelo DSPN, considerando-se, porém, as dependências de marcação [86]:

$$\text{EDSPN}=(P,T,I(\cdot),O(\cdot),H(\cdot),M_0,D(\cdot),W(\cdot))$$

Onde:

- um conjunto de m lugares $\mathbf{P}=(p_1, p_2, \dots, p_m)$;
- um conjunto de n transições imediatas, com tempo de disparo zero, e transições exponenciais e determinísticas, com tempo de disparo diferente de zero.
 $\mathbf{T}=(t_1, t_2, \dots, t_n)= T_{im} \cup T_{exp} \cup T_{det}$;

Considerando um lugar p_i e uma transição t_j , as multiplicidades dos correspondentes arcos de entrada, arcos de saída e arco inibidor, denotadas respectivamente por $i_i(t_j)$, $o_i(t_j)$ e $h_i(t_j)$, dependentes da marcação, são:

$$\begin{aligned} \forall p_i \in \mathbf{P}, \forall t_j \in \mathbf{T}: i_i(t_j): \mathbf{P} \times \mathbf{T} \times \mathbf{N}^{|\mathbf{P}|} &\rightarrow \mathbf{N}, \\ o_i(t_j): \mathbf{T} \times \mathbf{P} \times \mathbf{N}^{|\mathbf{P}|} &\rightarrow \mathbf{N}, \\ h_i(t_j): \mathbf{T} \times \mathbf{P} \times \mathbf{N}^{|\mathbf{P}|} &\rightarrow \mathbf{N}, \end{aligned} \quad (2.70)$$

- uma função de prioridade de disparo $\mathbf{\Pi}(\cdot)$ para transições imediatas (T_{im}), i.e., $\forall t_j \in T_{im}: \mathbf{\Pi}: T_{im} \rightarrow \mathbf{N}$;
- uma marcação inicial $\mathbf{M}_0=(m_{01}, m_{02}, \dots, m_{0m})$;
- uma função de retardo de disparo $\mathbf{D}(\cdot)$ para transições temporizadas. Para T_{exp} (exponencial) especifica o tempo médio de disparo, enquanto para T_{det} (determinístico) especifica um retardo de disparo constante, os quais não precisam ser os mesmos para todas as marcações nas quais a transição temporizada está habilitada. Formalmente, a função de retardo de disparo $\mathbf{D}(\cdot)$ em função da marcação é dada por:

$$\forall t_j \in (T_{exp} \cup T_{det}): \mathbf{D}(t_j): (T_{exp} \cup T_{det}) \times \mathbf{N}^{|\mathbf{P}|} \rightarrow (0, \infty) \quad (2.71)$$

- uma função peso de disparo $\mathbf{W}(\cdot)$ associada com transições imediatas (T_{im}), a qual pode ser uma função da marcação usada para decidir a transição imediata que deve disparar quando mais de uma transição imediata, com a mesma prioridade, forem habilitadas concorrentemente, é dada por:

$$\forall t_j \in T_{im}: \mathbf{W}(t_j): T_{im} \times \mathbf{N}^{|\mathbf{P}|} \rightarrow (0, \infty) \quad (2.72)$$

EDSPN tem uma representação mais compacta que DSPN e não afeta a análise do processo estocástico. Para uma maior compreensão das extensões utilizadas nos modelos EDSPN, algumas empregadas com ênfase nos modelos desta Tese, as seguintes definições [140][60][22][38][100] são apresentadas:

- **multiplicidade dos arcos:** esta extensão associa aos arcos de entrada ou saída das transições um peso maior do que 1. Pode-se representar a multiplicidade dos arcos como um conjunto de arcos paralelos. Esta extensão permite a remoção de *tokens* dos lugares de entrada ou o depósito de *tokens* nos lugares de saída em quantidades maiores do que 1 quando uma determinada transição dispara;
- **função de guarda:** é um predicado geral que determina quando uma transição estará habilitada. É uma restrição a mais imposta sobre a transição imediata, em função da

marcação corrente, em adição aos arcos de entrada, as prioridades e aos arcos inibidores. Somente será verificada quando as demais restrições não forem satisfeitas. É uma característica poderosa que simplifica a representação gráfica e a torna mais fácil de ser compreendida. É bastante utilizada nos modelos EDSPN para modelagem dos blocos que compõem um sistema, a serem apresentados nos próximos capítulos. Esta extensão mostra-se bastante interessante na modelagem dos blocos formados conjuntamente por hardware e software;

- **multiplicidade dos arcos dependentes da marcação:** esta extensão pode ser aplicada quando o número de *tokens* transferidos de um lugar ou para um lugar dependerem da marcação corrente. Um modo bastante comum é a sua utilização na eliminação de todos os *tokens* de uma só vez após um único disparo da transição. É utilizado nos modelos MDP, a serem vistos nos próximos capítulos, durante análise numérica associadas às diversas amostras do tempo;
- **taxas de disparo dependentes da marcação:** esta característica permite que taxas de disparo das transições temporizadas possam ser especificadas em função da marcação corrente. Por exemplo, estratégias de reparo podem ser definidas em função do número de *tokens* nos lugares de falha dos modelos correspondentes aos blocos.

- **multiplicidade de arco condicional dependente da marcação:** esta característica é suportada pela ferramenta de modelagem utilizada nos modelos que foram desenvolvidos nesse trabalho. Consiste na elaboração de uma série de multiplicidades de arcos dependentes das marcações correntes, em especial de lugares que apresentam condições Booleanas, denominados *flags*. Por meio dessa característica diferentes condições de multiplicidade estão contidas numa mesma expressão. A expressão de multiplicidade condicional é formada por um conjunto de construtores IF's seguidos por : (condição THEN) e por um construtor ELSE ao final da expressão, conforme exemplo a seguir:

$$IF\#Pi = 1\ AND\ \#Rel_Flag = 1 : 1IF(\#Pi > 1\ AND\ \#Pi \leq 3)\ AND\ \#Rel_Flag = 1 : 2ELSE3 \quad (2.73)$$

- **taxas de disparo condicional dependentes da marcação:** esta característica também é suportada pela ferramenta de modelagem e utilizada freqüentemente em um grande número de modelos dessa Tese. Consiste na definição de diversas taxas de disparo dependentes das marcações correntes, em especial dos *flags*. Por meio dessa característica, diferentes taxas de disparo são colocadas numa mesma expressão. São representadas por um conjunto de construtores IF's seguidos por : (condição THEN) e por um construtor ELSE ao final da expressão, conforme exemplo a seguir:

$$IF\#Pi = 1 : (MTTF)IF\#Pi > 1\ AND\ \#Pi \leq 3 : (2 * MTTF)ELSE(MTTF / 2) \quad (2.74)$$

A sintaxe correspondente pode ser observada no ANEXO-B. Caso o leitor necessite de mais informações, poderá consultar [134]. Por meio de expressões lógicas condicionais das multiplicidades de arcos e das taxas de disparo, os modelos desse trabalho se tornam bastante concisos e permitem que situações diversas possam ser contempladas numa mesma expressão.

Para transformar o modelo EDSMN no correspondente processo estocástico de nível de abstração mais baixo, de modo a possibilitar a análise propriamente dita do modelo, são necessárias duas tarefas: a) uma política de disparo das transições através da definição de políticas de execução para o modelo, com respeito a regras de escolha da próxima transição a disparar em uma determinada marcação; b) uma política de memória por meio de critérios que levem em conta o passado do modelo com relação às temporizações das transições do modelo.

As políticas de disparo associadas às transições temporizadas podem ser executadas de dois modos distintos:

- **política de corrida (*race policy*):** quando várias transições temporizadas estiverem habilitadas em uma dada marcação, aquela que tiver o menor retardo associado será aquela que irá disparar primeiro;
- **adoção de métrica específica,** por meio, por exemplo, da condição de um *flag*.

As políticas de disparo associadas às transições imediatas também podem ser executadas de dois modos distintos:

- **associação de mecanismo de prioridade:** o disparo das transições imediatas será função do nível de prioridade associado às mesmas. Disparará primeiro aquela transição que tiver um maior nível de prioridade. Esta escolha é determinística;
- **associação de uma função de probabilidade discreta:** um conjunto de probabilidades discretas é associado às transições imediatas em conflito. Neste caso a escolha da transição imediata a ser disparada é feita de forma aleatória na frequência da probabilidade constante associada. Este tipo de associação é estático;
- **associação de função de distribuição de probabilidade:** esta forma de política de disparo suportada pela ferramenta de modelagem é introduzida nesta Tese como forma de permitir o disparo de transições imediatas em conflito por meio de expressões analíticas das funções de distribuição de probabilidade. Ao invés de um valor de probabilidade discreto são definidas expressões analíticas, do tipo confiabilidade e inconfiabilidade, disponibilidade e indisponibilidade, detecção de falha e não detecção de falha, cuja soma é igual a 1. Por exemplo, ao se executar uma determinada atividade cuja função de distribuição seja exponencial, esta pode resultar em um valor confiável, representado por $(\exp^{-\lambda t})$, ou um valor inconfiável da operação, representado por $(1 - \exp^{-\lambda t})$. Este tipo de associação dinâmica depende dos parâmetros contidos na expressão.

Com relação às políticas de memória, duas alternativas básicas são possíveis na mudança de marcação[94][95]:

- **continue:** as temporizações associadas às transições mantêm os valores presentes que continuarão a serem decrementados nas próximas marcações;

- **restart**: as temporizações associadas às transições são reiniciadas, ou seja, os valores presentes são descartados e novos valores serão gerados quando necessário.

Diferentes comportamentos provenientes dos sistemas reais definem diferentes combinações dos mecanismos *continue* e *restart* para as transições que não puderam disparar:

- **resampling**: após o disparo de uma transição, as temporizações de todas as transições são descartadas. Novos valores são então estabelecidos para as transições habilitadas na nova marcação;
- **enabling memory**: após o disparo de uma transição, as transições que permanecerem habilitadas terão as suas temporizações correntes preservadas, enquanto que as transições que não permanecerem habilitadas serão reiniciadas para um novo valor;
- **age memory**: após o disparo de uma transição, as demais transições, habilitadas ou não, terão os seus valores correntes de temporização preservados.

Quanto à semântica de temporização, as transições temporizadas com grau de habilitação maior do que 1 podem ser caracterizadas da seguinte forma:

- **semântica de servidor único (*single server*)**: quando mais de um *token* está presente em um determinado lugar, um retardo de disparo é associado à transição quando esta é inicialmente habilitada, e novos retardos são gerados após o disparo da transição se a transição se mantém ainda habilitada na nova marcação. Neste tipo de semântica de temporização os *tokens* são processados serialmente. Este tipo de semântica é utilizada nos modelos de disponibilidade, considerando-se que haja apenas uma única equipe de reparo, quando vários blocos que compõem o sistema entram numa condição de falha;
- **semântica de múltiplos servidores (*multiple server*)**: uma quantidade de *tokens* é processada até um grau máximo K de paralelismo. Caso o grau de habilitação seja maior do que K , as temporizações associadas aos novos *tokens* ainda não processadas serão ativadas, quando o número de temporizações concorrentes sendo processadas cair abaixo do valor K . A especificação temporal completa deste tipo de semântica depende do grau de habilitação, isto é, do número de *tokens* e do número de servidores K . Este tipo de semântica é utilizado nos modelos de disponibilidade considerando-se que haja um número de reparadores menor do que o número de blocos na condição de falha. Os blocos em excesso ficarão em fila;
- **semântica de infinitos servidores (*infinite server*)**: neste caso, sendo o valor de K infinito, todos os *tokens* serão processados em paralelo e as temporizações a eles associadas serão decrementadas a zero em paralelo. Este tipo de semântica é utilizada nos modelos de disponibilidade considerando-se que haja tantas equipes de manutenção quantos sejam os blocos em falha. Para cada bloco existe uma equipe de

manutenção exclusiva e independente. Neste tipo de semântica todos os *tokens* são processados em paralelo.

2.3.1 Propriedades Comportamentais das Redes de Petri

Para que se possa verificar o correto funcionamento de um modelo é necessário uma análise qualitativa de suas propriedades. Algumas das mais importantes, descritas de um modo mais detalhado em [128][94][43][60][61], serão apresentadas, de um modo sucinto e informal, a seguir:

- **alcançabilidade (*reachability*):** uma marcação M' é alcançável a partir de uma marcação M se e somente se existir uma seqüência de disparo de transições capazes de conduzir a marcação M à marcação M' . Alcançabilidade define a possibilidade do modelo do sistema se encontrar em uma dada marcação M ;
- **reversibilidade (*reversibility*):** uma rede de Petri é dita reversível se e somente se de qualquer estado alcançável a partir da marcação inicial M_0 , é possível se retornar a M_0 . Reversibilidade expressa a possibilidade de uma rede de Petri retornar a uma determinada marcação freqüente e infinitamente. Esta marcação, a qual poderá ser a marcação inicial ou qualquer outra marcação, é denominada *home state*. Quando a rede de Petri é reversível, o grafo de alcançabilidade é fortemente conectado;
- **ausência de *deadlock* (*deadlock free*):** uma rede de Petri contém *deadlock* quando atinge um estado no qual nenhuma transição é possível ser disparada. Esta é uma das mais importantes propriedades das redes de Petri uma vez que ela define possibilidades de travamento do sistema;
- ***liveness*:** uma transição é dita viva em uma rede de Petri se e somente se para cada marcação M alcançável de M_0 , existir uma marcação M' , alcançável de M de modo que a transição esteja habilitada. Ou seja, uma transição é viva se em cada marcação alcançável uma nova marcação possa ser alcançada na qual a transição esteja habilitada. Uma rede de Petri é viva se todas as transições são vivas;
- **limitação (*boundedness*):** uma rede de Petri é dita ser k -limitada se e somente se, para cada marcação alcançável M , o número de *tokens* nesse lugar é menor do que ou igual a um limiar k . A consequência desta propriedade está na natureza finita do espaço de estados;
- **segurança (*safeness*):** uma rede de Petri é dita segura quando o fator de limitação k da rede k -limitada for igual a 1.
- **exclusão Mútua (*Mutual Exclusion*):** esta propriedade lida com a impossibilidade da simultânea marcação de dois lugares, ou ainda com a habilitação simultânea de duas transições em uma marcação.

O conjunto de todas as possíveis marcações de uma rede de Petri alcançáveis por meio de uma seqüência de disparos de transições, a partir de uma marcação inicial, é denominado conjunto de alcançabilidade *RS*. As marcações que compõem o grafo de alcançabilidade são particionadas em dois conjuntos: o conjunto das marcações não tangíveis, ou *vanish*, e o conjunto das marcações tangíveis, ou *tangible*. As marcações não tangíveis são aquelas compostas por pelo menos uma transição imediata, enquanto as marcações tangíveis são aquelas compostas apenas por transições temporizadas exponenciais. Conflitos entre transições imediatas, em uma marcação *vanish*, são resolvidos através do uso de escolha aleatória conforme as probabilidades associadas aos pesos das transições imediatas em conflito. O conjunto de marcações tangíveis de uma rede de Petri estocástica é isomórfica com a CTMC correspondente o que facilita o processo de geração e solução das cadeias de Markov, por meio dessas redes. A explosão de estados ocorre quando o conjunto de marcações ou estados, em um conjunto de alcançabilidade, é infinito ou extremamente elevado, o que impossibilita avaliações analíticas ou numéricas.

2.3.2 Técnica de Análise Qualitativa por Regras de Redução Simples

Uma técnica usual para análise qualitativa é a solução baseada em transformação, uma vez que a análise de redes de grandes dimensões não é uma tarefa trivial. Por isso, a disponibilidade de métodos que permitam a transformação de modelos, por meio de redução, mantendo as propriedades qualitativas, tem sido estudada. As técnicas de redução são baseadas na transformação do modelo original em um modelo abstrato, de modo que as propriedades de *liveness*, limitação e *safeness*, sejam preservadas nos modelos reduzidos. Algumas regras simples de redução [142], comumente utilizadas, são apresentadas na Figura 2.3.

As regras de redução mostradas na Figura 2.3, são:

- (a) fusão de lugares seriais;
- (b) fusão de transições seriais;
- (c) fusão de lugares paralelos;
- (d) fusão de transições paralelas;
- (e) eliminação de lugares de auto-loop;
- (f) eliminação de transições de auto-loop;

Estas regras efetuam a transformação das redes por meio de fusões de lugares e de transições, além da eliminação de *loops*, preservando as propriedades citadas anteriormente.

2.3.3 Síntese de redes de Petri

O objetivo desta seção é apresentar, de modo simplificado, as técnicas *bottom-up*, *top-down* e híbrida para síntese de redes de Petri. Esta seção está relacionada aos aspectos de geração do espaço de estados, quando se utiliza redes de Petri para modelagem de sistemas muito complexos. Utilizar apenas métodos de redução para análise das propriedades das redes de Petri pode não ser suficiente, uma vez que os métodos de redução não são eficientes para sistemas que têm muitos recursos compartilhados. Uma solução alternativa é a utilização de métodos de modelagem sistemática que garanta as propriedades qualitativas de projeto. Estes métodos de síntese objetivam evitar a explosão de estados para sistemas complexos e eliminar a necessidade de análise. Informações mais detalhadas podem ser obtidas em [45].

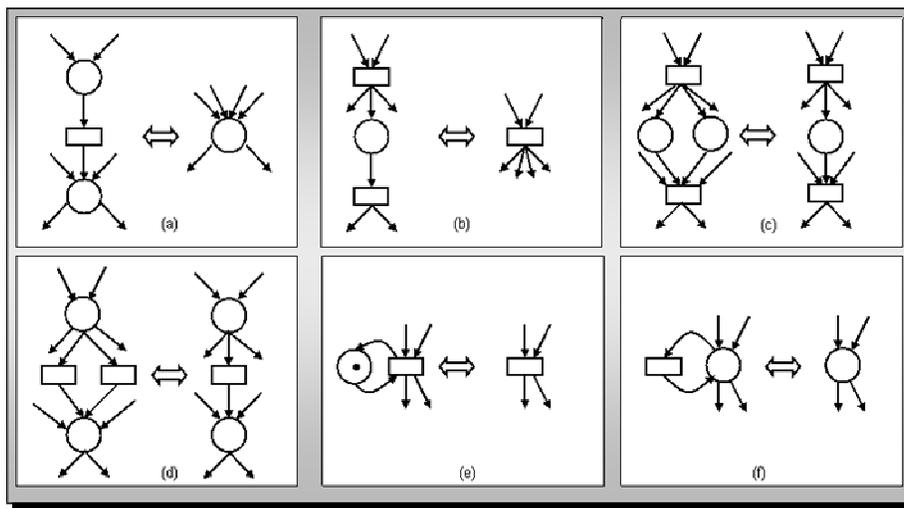


Figura 2.3 Regras de redução

2.3.3.1 Método de Composição Modular ou Síntese *Bottom-up*

Método de composição modular ou *bottom-up* é um dos métodos comumente utilizado no processo de modelagem. Este método envolve a especificação do sistema em subsistemas (blocos) menores e algum procedimento sistemático para combinar estes blocos, com suas interações, em um sistema integrado. Os blocos, em geral, são muitos simples e de fácil verificação. Algumas interações são representadas por lugares, transições ou caminhos comuns, dos blocos individuais, as quais são consideradas em cada etapa da síntese. Os blocos envolvidos são combinados através da junção destes lugares e/ou transições em um subsistema. A análise do subsistema é realizada logo após cada etapa de síntese, como forma de simplificar a análise do sistema como um todo ao final do processo, quando além do sistema, algumas propriedades importantes são obtidas. Através da Figura 2.4 é mostrada uma nova rede, pela junção de duas outras por meio da interação de lugar comum, conforme provado por meio de teorema em [8][45].

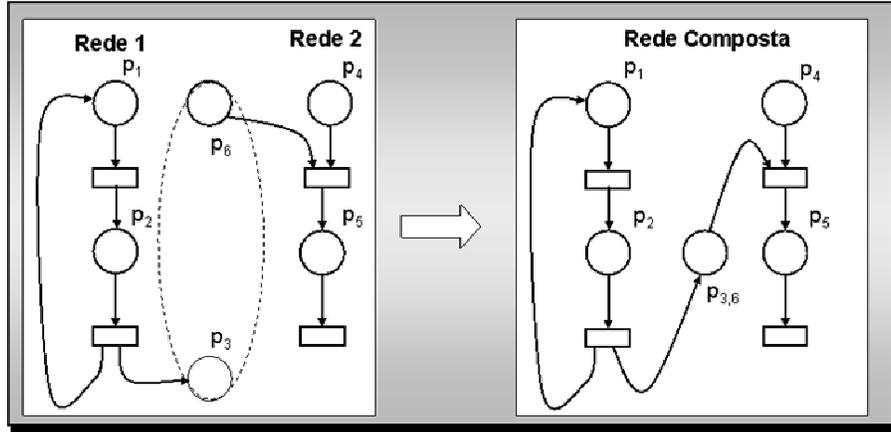


Figura 2.4 Junção de um conjunto de lugares (1-Way Merge)

Em cada etapa da síntese, segundo o método de Agerwala and Choed-Amphai [1], sub-redes podem ser formadas pela junção de um conjunto de lugares em um novo lugar. Isto é denominado *1-way merge*, cujo formalismo será descrito a seguir.

Dada uma rede de Petri EDSPN $N=(P,T,I,O)$, formada apenas por lugares, transições, arcos de entrada e de saída, deve ser selecionado um conjunto de lugares a serem juntados, por meio da fusão de lugares seriais ou paralelos, em $P_m \subseteq P$, tal que:

1. Para qualquer $p_i, p_j \in P_m$, se $(p_i, t) \in I$ e $(p_j, t) \in I$, logo $i=j$;
2. Para qualquer $p_i, p_j \in P_m$, se $(p_i, t) \in O$ e $(p_j, t) \in O$, logo $i=j$;

Logo, a nova rede gerada $N'=(P',T',I',O')$ será tal que:

1. $T'=T$; (2.75)
2. $P'=(P-P_m) \cup \{p\}$ onde $p \notin P$;
3. I' e O' são obtidas pela troca de cada ocorrência de $p_i \in P_m$ em I e O por p ;

Isto pode ser exemplificado pela rede de Petri da Figura 2.4, $N=(P,T,I,O)$, formada pelos seguintes elementos:

- P - conjunto de lugares de entrada e de saída das duas redes;
- T - conjunto de transições de entrada e de saída das duas redes;
- I - conjunto de arcos de entrada das duas redes;
- O - conjunto de arcos de saída das duas redes.

Observa-se que o conjunto $\{p_3, p_6\}$ representa os lugares a serem juntados para geração do lugar $p_{3,6}$ em uma nova rede de Petri $N'=(P', T', I', O')$, de tal modo que:

$$P' = (P - \{p_3, p_6\}) \cup \{p_{3,6}\} \quad (2.76)$$

2.3.3.2 Síntese *Top-down*

Numa implementação *top-down*, o modelo agregado é iniciado em um nível de abstração elevado e aos poucos, passo a passo, vai sendo refinado, ou seja, vai incorporando detalhes, que o conduzem ao estágio final. A incorporação de detalhes poderá ser feita por expansão de lugares e expansão de transições. O refinamento prosseguirá até que seja atingido o grau de detalhamento exigido pelo sistema. Uma das vantagens desta técnica está na visão global do sistema do início ao fim da síntese. Em [142] métodos são fornecidos para garantir que a cada etapa da síntese propriedades importantes do sistema não sejam perdidas, o que torna desnecessária a análise final do sistema.

2.3.3.3 Síntese Híbrida

Esta técnica de síntese faz uso dos dois métodos de síntese aplicados às redes de Petri vistas anteriormente. Quando necessário um ou outro método poderá ser usado de modo alternado. Deste modo, a complexidade do detalhamento do problema é aliviada. De modo a evitar a análise qualitativa para os sistemas complexos, a metodologia inclui um conjunto de estruturas de exclusão mútua. A utilização deste método em sistemas de manufatura poderá ser obtida em [45].

A utilização do formalismo EDSPN na descrição dos modelos e no processo de análise e modelagem dos sistemas dependáveis faz uso de mecanismos de tolerância a falhas, por meio de diversos tipos de redundância, de modo a satisfazer os requisitos de dependabilidade dos sistemas. Deste modo, para que haja uma melhor compreensão da taxonomia e dos conceitos de dependabilidade e tolerância a falhas empregados nesta Tese, se faz necessário a sua descrição.

2.4 Dependabilidade e Tolerância a Falhas

Define-se dependabilidade [80], segurança de funcionamento, ou ainda confiança no funcionamento [69], como a qualidade do serviço liberado, tal que confiança possa ser justificadamente depositada neste serviço. Esta qualidade será avaliada quantitativamente nesta Tese, através dos critérios de confiabilidade, disponibilidade e segurança, os quais associados aos critérios de manutenibilidade, segurança contra intrusão, integridade e confidencialidade, definem os atributos de um sistema dependável [13], conforme a Figura 1.1.

2.4.1 Principais Conceitos e Taxonomia

Através da Figura 1.1 pode-se observar que a dependabilidade de um sistema computacional sofre ameaças através das falhas, erros e defeitos. As falhas podem ser caracterizadas por meio de alguns critérios significativos [30][80][13][58], os quais são mostrados de um modo sucinto na Figura 2.5.

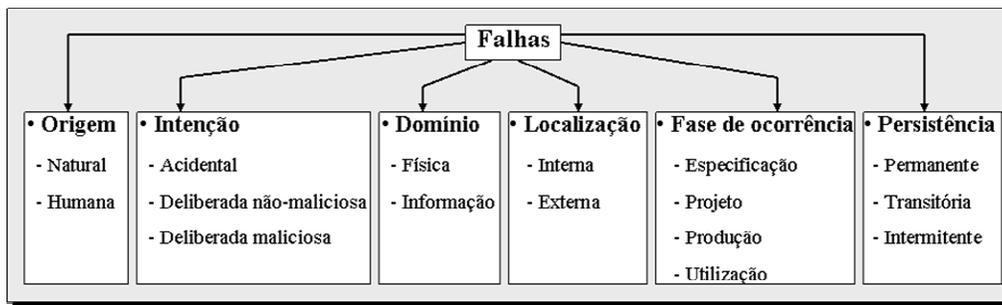


Figura 2.5 Definição das classes de falhas

As falhas devidas a componentes de hardware são de natureza física, enquanto as falhas dos componentes de software de um sistema são devido a erros na especificação, projeto e implementação, uma vez que os componentes de software não sofrem mudanças funcionais devido a interações externas ou envelhecimento [68]. A falha é dita latente, quando não produz um erro e é dita ativa, quando um erro é produzido como consequência de sua ação, por meio de desvio do estado desejado, pela violação do valor atribuído ao estado ou da transição entre estados. As diversas classificações de erro [80][13][58] estão relatadas de um modo conciso na Figura 2.6.

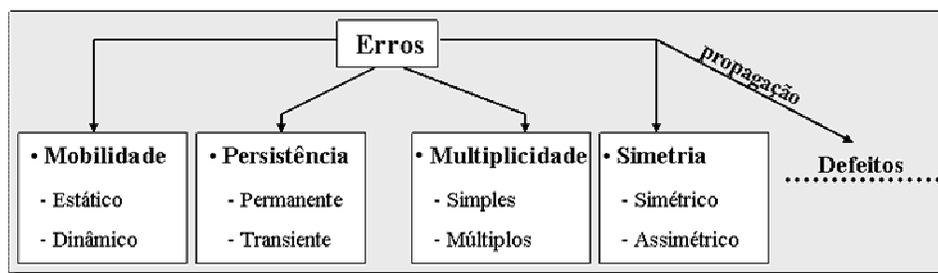


Figura 2.6 Classificação de erros

A propagação do erro dentro de um sistema conduz a um defeito. As possibilidades de erros que conduzem a defeitos do sistema são funções dos diferentes modos de falha. Defeitos são gerados quando o serviço liberado não está de acordo com o serviço que foi estabelecido durante a especificação. Defeitos podem ser caracterizados, de acordo com as diversas possibilidades de falha do produto, no que se conceitua chamar modos de falha (*failure modes*). As diferentes ocorrências de defeitos, que geram serviços incorretos, podem ser caracterizadas de acordo com a Figura 2.7.

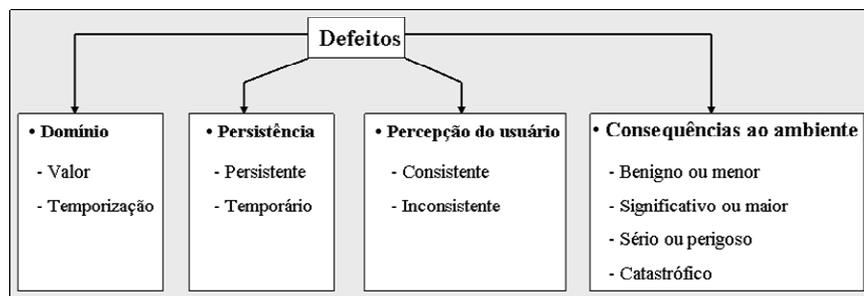


Figura 2.7 Classificação das ocorrências de defeito.

Falhas, erros e defeitos são mecanismos destrutivos que tentam impedir o correto funcionamento de um sistema em função de uma sucessão de eventos indesejáveis [58]. Meios de proteção são estabelecidos para evitar que tais mecanismos alterem o comportamento de um sistema, ou ainda, que o sistema possa liberar o serviço especificado mesmo na presença desses mecanismos. Várias são as maneiras de se representar os diferentes meios para obtenção de dependabilidade.

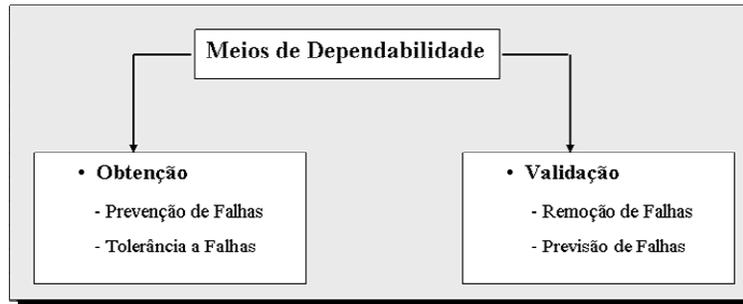


Figura 2.8 Meios para obtenção e validação da dependabilidade de um sistema

A representação descrita em [68] é particularmente interessante, pois divide estes meios, quanto à sua obtenção e quanto à sua validação, conforme mostrado na Figura 2.8.

2.4.2 Meios de Validação de Dependabilidade

- I. **remoção de falhas:** é realizada durante as fases de desenvolvimento (especificação e design), produção e operacional de um sistema. Para cada fase, diferentes técnicas poderão ser utilizadas. O processo de remoção de falhas durante a fase de desenvolvimento de um sistema consiste em três etapas distintas: Detecção de falhas, localização de falhas e remoção de falhas. As formas de diagnosticar as falhas podem ser do tipo diagnóstico retrospectivo, cujo objetivo é a determinação das causas que levaram o sistema a um defeito, ou diagnóstico preditivo, cujo objetivo é determinar quando a falha ocorrerá [126]. A remoção de falhas durante a operação de um sistema poderá ser realizada por meio de manutenção corretiva, preventiva e evolutiva [80][58]. Quanto a troca dos componentes defeituosos por componentes novos, a manutenção de equipamentos eletrônicos, em especial sistemas computacionais ou de telecomunicações, poderão ser do tipo reparo a quente, sem a necessidade de parada do equipamento ou reparo a frio, quando o sistema é colocado fora de serviço.
- II. **previsão de falhas:** são técnicas utilizadas para avaliação do comportamento do sistema com respeito a ocorrência de falhas ou ativação. As avaliações podem ser de dois tipos:
 - **avaliações qualitativas:** cujo objetivo é identificar, classificar, escalonar os modos de falha, ou a combinação de eventos que conduziria o sistema a defeitos. Os métodos usados para este tipo de avaliação são os modos de falha e análise de efeito;
 - **avaliações quantitativas:** cujo objetivo é avaliar em termos probabilísticos o grau em que os atributos de dependabilidade de um sistema, são satisfeitos.

Estes atributos são definidos como as medidas de dependabilidade do sistema. Os métodos usados podem ser cadeias de Markov e redes de Petri estocásticas, no caso desta Tese, EDSPN.

Caso deseje-se realizar os dois tipos de análises através de um único método, utilizam-se os diagramas de blocos de confiabilidade, as árvores de falha, ou as redes de Petri estocásticas.

2.4.3 Meios de Obtenção de Dependabilidade

- I. **prevenção de falhas:** os meios de prevenção de falhas estão associados a técnicas de controle de qualidade, empregadas durante as diversas fases de um sistema de hardware e software, de modo a evitar ou reduzir a ocorrência de falhas. A modularização dos sistemas de hardware e software, a blindagem contra interferências eletromagnéticas e o isolamento térmico contra elevação de temperatura, além de uma interface amigável com o usuário, são algumas das técnicas utilizadas;
- II. **tolerância a falhas:** esta abordagem, mais realista, parte do pressuposto que falhas ocorrerão apesar de todas as medidas preventivas que vierem a ser tomadas, e que o sistema deve liberar um serviço correto mesmo na presença de falhas [6]. É geralmente implementada por detecção de erro e subsequente recuperação do sistema.
 - As detecções de erro que originam uma mensagem ou um sinal dentro do sistema podem ser de duas classes distintas:
 - i. **detecção de erros concorrentes**, ocorre em paralelo à liberação do serviço;
 - ii. **detecção de erros preemptivas**, ocorrem com a interrupção da liberação do serviço;
 - O processo de recuperação restaura o sistema afetado por erros e falhas para um estado livre destas condições indesejáveis. A recuperação pode ser por manipulação de erros (*error handling*), de falhas (*fault handling*) ou *fault masking*:
 - i. ***error handling***: elimina erros do sistema por meio de retorno (*rollback*) a um estado armazenado antes da ocorrência do erro, denominado *checkpoint*, ou por meio de avanço (*rollforward*) a um novo estado;
 - ii. ***fault handling***: evita que falhas localizadas possam ser novamente ativadas. Esta solução envolve quatro passos: a) localização ou diagnóstico da falha (localização da falha e tipo); b) isolamento da falha por meio de exclusão física ou lógica dos componentes em falha; c) reconfiguração do sistema pela comutação de componentes redundantes ou redistribuição de tarefas entre os componentes ativos; d) reinicialização do sistema que atualiza a nova configuração do sistema e atualiza tabelas e registros.

- iii. ***fault masking*** (mascaramento de falha): a recuperação neste caso ocorre quando o número de componentes redundantes é suficiente para permitir a recuperação, sem detecção explícita do erro.

Apesar de todas as precauções estabelecidas nas estratégias anteriores, falhas ainda permanecerão no sistema, seja pelo fato de não terem sido devidamente detectadas e tratadas, seja pelo fato de terem surgido naturalmente no sistema. Diferentemente dos meios de prevenção de falhas e de remoção de falhas, a estratégia de tolerância a falhas age sobre os efeitos e não sobre as causas. Sistemas tolerantes a falhas possibilitam uma significativa redução de custos de manutenção pelo aumento da disponibilidade do sistema. Tolerância a falhas está baseada nas técnicas de redundância e tem por finalidade evitar o aparecimento de defeitos, e não o de corrigir as causas dos erros. Os mecanismos de tolerância a falhas conseguem incrementar a confiabilidade de um sistema até um nível desejado quando atuam em conjunto com as técnicas de prevenção, predição e remoção de falhas [78]. A incorporação de elementos redundantes a um sistema pode ocorrer de várias formas [70]:

- **redundância de hardware:** consiste na replicação dos recursos de hardware de um sistema;
- **redundância de software:** consiste na utilização de variantes [81][82] das rotinas de software de um sistema, as quais devem obedecer a uma mesma especificação;
- **redundância de informação:** consiste na adição de bits extras de informação além daqueles requeridos para a implementação de uma dada função;
- **redundância temporal:** consiste na adição de um tempo adicional para a execução das funções de um sistema de modo permitir a detecção de falhas;

As técnicas de redundância utilizadas para dispositivos de hardware podem ser dos seguintes tipos:

- **redundância estática, ou passiva:** realiza tolerância a falhas sem a necessidade de detecção de qualquer falha. Para isso utiliza módulos extras de hardware com o objetivo de mascarar o efeito dos módulos falhos;
- **redundância dinâmica, ou ativa:** requer ações de detecção e recuperação de falhas, por meio da troca do módulo falho pelo módulo redundante em espera.
- **redundância híbrida:** combina as soluções de redundância estática e dinâmica nas ações de detecção e recuperação de falhas.

2.4.4 Atributos de Dependabilidade

Atributos são critérios de dependabilidade de um sistema de modo que confiança possa ser justificadamente depositada nos serviços liberados por este sistema. Os atributos de um sistema dependável são os seguintes:

- **confiabilidade (*Reliability*):** é a probabilidade condicional que um sistema tenha sobrevivido em um ambiente especificado até um tempo t , dado que ele estava

- operacional em um tempo 0 [58]. Pode-se definir ainda como a continuidade do serviço correto de um sistema [80];
- **disponibilidade (*Availability*)**: é a probabilidade que um sistema funcione corretamente em um tempo t , sabendo-se que ele estava em funcionamento correto no tempo inicial [58]. É ainda definido como a prontidão do sistema para um serviço correto [80];
 - **segurança (*Safety*)**: é a probabilidade de ausência de falhas do sistema cujas conseqüências externas sejam catastróficas [58]. Pode-se definir ainda como a ausência de conseqüências catastróficas sobre os usuários e o ambiente externo ao sistema [80];
 - **manutenibilidade (*Maintainability*)**: é uma medida da habilidade do sistema de submeter-se a reparos, pela supressão da falha presente, e de permitir modificações, pela adição de novas funcionalidades ou pela melhoria das já existentes [80][58];
 - **segurança contra intrusão (*Security*)**: é uma medida da prevenção de acesso não autorizado e/ou manipulação da informação [80][58];
 - **confidencialidade (*Confidentiality*)**: é uma medida da ausência de divulgação não autorizada da informação [80][58];
 - **integridade (*Integrity*)**: é uma medida da não ocorrência de alterações impróprias da informação [58], ou segundo [80], é uma medida da ausência de alterações impróprias do estado do sistema.

2.5 Técnicas de Redundância de Hardware

As técnicas de redundância do hardware tanto podem ser estáticas (passivas), quanto dinâmicas (ativas). Os mecanismos de tolerância a falhas associados a estas técnicas serão descritos, de um modo conciso, a seguir.

2.5.1 Técnicas de Redundância Estática ou Passiva

As técnicas de redundância estática se baseiam nos mecanismos de votação para mascaramento da ocorrência de falhas. Nesse tipo de redundância não há necessidade de mecanismos de detecção do erro. A maior parte das soluções emprega o conceito de votação de maioria. Algumas técnicas que utilizam este conceito serão apresentadas.

- **Redundância Modular Tripla (TMR)**

Originalmente proposta por Von Veumann [143][13], este modelo é apresentado na Figura 2.9. Esta técnica inibe o efeito de falhas e evita o aparecimento de defeitos, por meio da utilização de módulos redundantes triplos. A saída desta técnica redundante é obtida por meio do voto majoritário das saídas dos módulos redundantes que a compõem [78]. Tal esquema produzirá um resultado correto, quando houver pelo menos dois módulos produzindo resultados corretos. Assim, um resultado errôneo em um desses módulos não terá efeito na saída final. A confiabilidade associada a técnica TMR será

função não apenas dos módulos triplicados, mas também do módulo votador. Portanto, a confiabilidade do esquema TMR não deverá ser melhor do que a confiabilidade do votador, pois isso implicaria em custos desnecessários. Vários exemplos de sistemas dependáveis utilizando a técnica redundante TMR podem ser encontrados em [126].

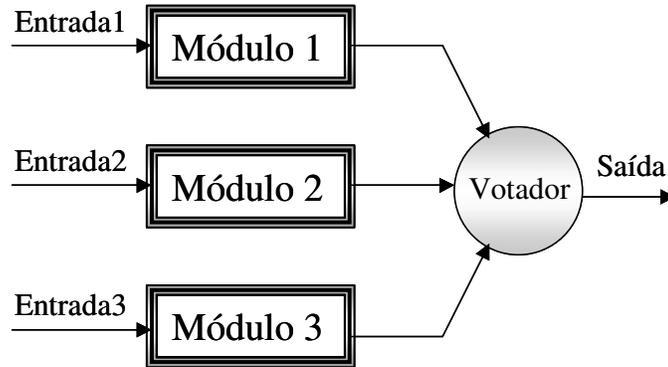


Figura 2.9 Redundância Modular Tripla (TMR)

De modo a evitar a utilização, por exemplo, de módulos de hardware de um mesmo fabricante e de um mesmo lote de produção que possam ocasionar o mesmo tipo de falha dando origem a um falso resultado correto, conhecido como falha de modo comum, é proposta uma diversidade na escolha dos módulos. Esta diversidade pode ser aplicada ao projeto dos módulos de hardware ou de software. Há um ganho significativo no uso da diversidade de projeto contra as falhas de modo comuns, e falhas do próprio projeto [97]. O uso de projeto diversitário para proteção dos sistemas contra falhas de modo comum, foi proposto em [10][77]. A idéia básica é que os modos de falha comum possam produzir diferentes erros por meio de diferentes implementações.

- **Redundância Modular Tripla Triplicado (TTMR)**

A confiabilidade do votador, ponto crítico de falha no sistema TMR básico, pode ser melhorada pela utilização de três módulos votadores idênticos (preferencialmente de projetos diversos). Este esquema denominado TMR Triplicado [78], ou TTMR, indica que dois dos três módulos produzirão um resultado correto, se e somente se, dois dos três módulos votadores funcionarem corretamente, conforme mostrado na Figura 2.10.

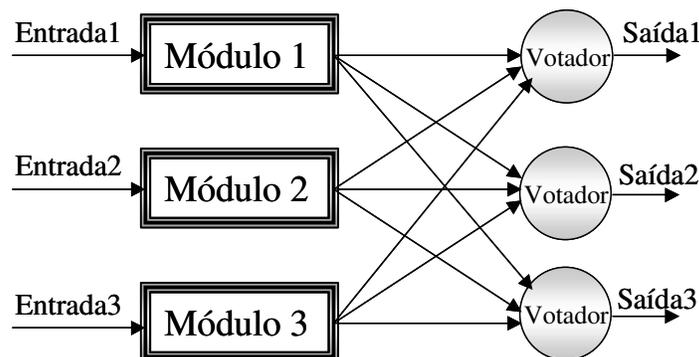


Figura 2.10 Redundância TMR Triplicada (TTMR)

O esquema TTMR produz três saídas corretas mesmo quando uma das entradas é falha. Essencialmente, este esquema restaura o sinal livre de erro na saída [70]. Este esquema poderá ser estendido, através da colocação de múltiplos estágios TTMR em cascata [70][126], de modo que a votação ocorra entre cada estágio, e que os erros sejam corrigidos antes de serem passados ao estágio seguinte, conforme pode ser observado na Figura 2.11.

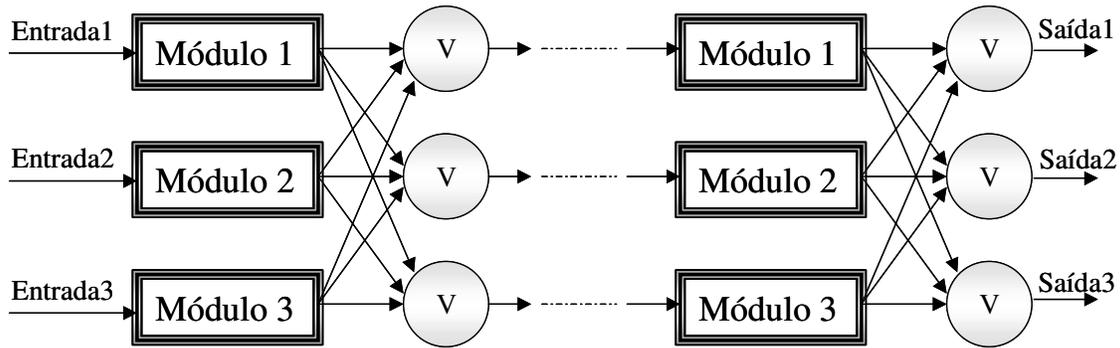


Figura 2.11 Múltiplos estágios TTMR

- **Redundância Modular de ordem N (NMR)**

Esta técnica de redundância estática propõe uma extensão do esquema TMR, pela multiplicidade de ordem N para os módulos redundantes, ao invés de somente três, como no caso TMR. Normalmente, N é definido como um número ímpar, de modo a se evitar um estado de incerteza na decisão do votador e permitir, conseqüentemente, um esquema de votação por maioria. Obviamente, o custo associado a esta técnica é N vezes o custo do módulo básico mais o custo do votador. Apesar de apresentar uma maior dependabilidade, este esquema tem uma perda de desempenho devido ao retardo causado pelo votador na propagação do sinal e da necessidade de sincronização dos N módulos [126]. O esquema NMR, mostrado na Figura 2.12, produz um resultado correto mesmo havendo (N-1)/2 módulos em falha. Logicamente, quanto mais módulos redundantes houver, maior será a tolerância do sistema a falhas, porém a um custo também proporcionalmente mais elevado. Em um sistema prático, o número de réplicas deve ser limitado de modo a reduzir o impacto no consumo de energia, no peso, no tamanho e no custo do sistema.

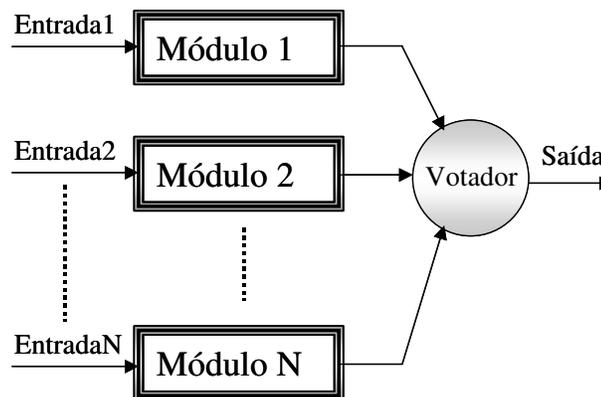


Figura 2.12 Redundância modular de ordem N

Os cálculos de confiabilidade e disponibilidade mostram que a técnica NMR é melhor do que a técnica TMR básica, porém a custo muito alto para um pequeno incremento nas medidas de dependabilidade [140]. Assim como no caso TMR, os sistemas NMR também podem ter o votador replicado, e a ligação em cascata de múltiplos estágios.

2.5.2 Técnicas de Redundância Dinâmica ou Ativa

As técnicas de redundância dinâmica ou ativa se baseiam nos mecanismos de detecção, localização e recuperação de falhas, ou erros [70]. Diferentemente da redundância passiva, este tipo de redundância não evita que as falhas geradas possam produzir erros no sistema. Redundância dinâmica é utilizada em aplicações que possuam redundância temporal, na qual um único módulo se encontra em operação em um dado instante de tempo, e as demais réplicas desse módulo se encontram em espera. Quando um erro é detectado no módulo que se encontra em operação, este é colocado fora de serviço, e uma das réplicas que se encontra em espera é colocada em operação, numa ação consecutiva de detecção e recuperação do erro. As técnicas utilizadas para detecção de erro, ou falha, podem ser englobadas em duas classes [126]:

- **detecção *off-line***: nesse tipo de detecção o módulo não é capaz de realizar qualquer tarefa útil enquanto estiver sob teste por programas de diagnóstico. Detecção por testes periódicos é uma das técnicas de detecção *off-line*. Falhas temporárias não são detectadas por esta técnica;
- **detecção *on-line***: nesse tipo de detecção a capacidade de checagem de falhas ou erros é realizada em tempo real pelo programa diagnóstico, paralelamente a realização das tarefas úteis pelo módulo sob teste. Detecção por auto-checagem e detecção por temporização de *watchdog*, são duas das técnicas de detecção *on-line* normalmente empregadas. A utilização de técnicas de redundância da informação por meio de CRC (código de redundância cíclica) ou checksum também permitem este tipo de detecção.

Na detecção por temporização do *watchdog*, devido a problemas econômicos, antes da comutação do módulo em falha para a condição fora de serviço deve-se verificar se a falha é transiente ou permanente por meio da técnica de *retry*. Por meio dessa técnica a operação é novamente executada. Se a falha persistir, o módulo é desativado e um módulo reserva, que realize a mesma função lógica, é ativado. Este processo, transparente ao usuário, de ativação e desativação de módulos que mantém a operação ininterrupta, é conhecido como auto-reparo (*self-repair*) [78].

- **Técnicas de Detecção de Falhas**
 - **Duplicação com Comparação**

Esta é uma das formas mais comuns de detecção, a qual utiliza dois módulos idênticos de hardware em paralelo para realização de uma mesma computação, conforme

pode ser observado na Figura 2.13. Os valores liberados pelos dois módulos são então comparados, e o resultado da comparação é então liberado. No caso de discordância, uma mensagem de erro é gerada; no caso de concordância uma mensagem válida é liberada. Esta é uma técnica de detecção e não de tolerância, uma vez que não se determina qual dos módulos está em falha. Esta técnica apresenta alguns problemas potenciais. Caso a entrada atribuída aos módulos seja inválida, ou caso a linha de comunicação que leva o valor de entrada aos módulos esteja em falha, ambos os módulos receberão valores idênticos e geração na saída idênticos valores, porém errôneos. Falhas no comparador produzirão a indicação de erro, quando este não existe, ou ainda produzirá um resultado válido pela impossibilidade de detecção de falha nos módulos duplicados [70].

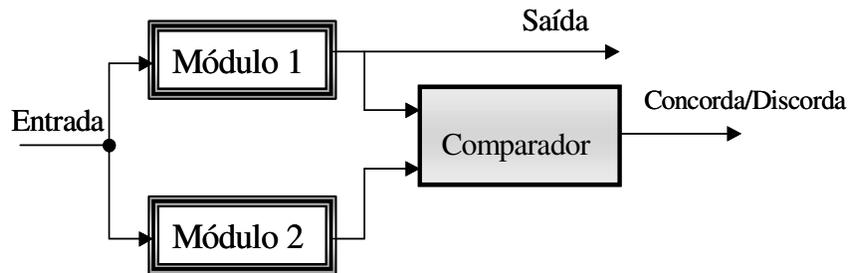


Figura 2.13 Duplicação com Comparação

- **Técnicas de Detecção e Reconfiguração Dinâmica do Hardware**

As técnicas de redundância dinâmica, nas quais um módulo se encontra ativo e os demais módulos se encontram numa posição de espera, aguardando a oportunidade de tornarem-se operacionais, na eventualidade de falha do módulo ativo corrente, são denominadas de técnicas de reserva em espera (*standby sparing technique*). Quando uma falha ocorre, o módulo faltoso ativo é colocado fora de serviço e comutado por um módulo passivo em espera [70][131].

A comutação do módulo ativo por um módulo passivo em espera, requer uma paralisação momentânea do sistema durante esse processo de reconfiguração. Levando-se em conta a paralisação, a técnica de redundância em espera pode ser de dois tipos: redundância de espera a quente (*hot standby sparing*) e redundância de espera a frio (*cold standby sparing*).

- **Técnica de Redundância de Espera a Quente (*Hot Standby Sparing*)**

Por meio da técnica de redundância de espera a quente, mostrada na Figura 2.14, os módulos reservas operam em sincronismo com o módulo ativo e estão prontos para sucedê-lo a qualquer instante [70][113][131]. Se as saídas de todos os módulos são as mesmas, a saída de qualquer módulo selecionado arbitrariamente poderá a qualquer instante ser a saída do sistema. Quando a técnica de redundância de espera a quente possui 2 módulos apenas, dá-se o nome a este arranjo de sistema *duplex*. Independentemente do tempo médio de execução dos dois módulos, o módulo reserva não deve liberar seu resultado até que o módulo primário se encontre numa condição de

falha. Caso as saídas dos módulos em um sistema *duplex* entrem em discordância, programas de diagnóstico entram em operação para localização da falha. Se a falha está ocorrendo em um dos módulos, este é colocado fora de serviço, e a operação do sistema se processará de modo *simplex* [77]. Nessa técnica de tolerância a falhas, tanto o módulo primário quanto os módulos reservas estão energizados e possuem uma mesma taxa de falha, a qual é constante e igual a λ .

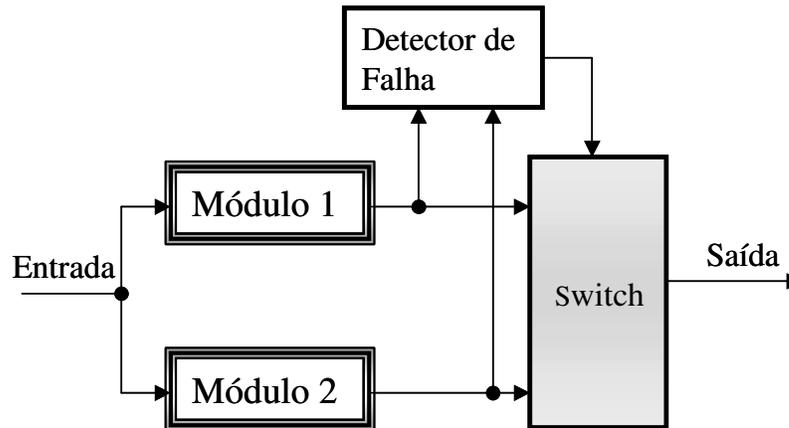


Figura 2.14 Módulo Redundante em Espera a Quente

- **Técnica de Módulo Reserva de Espera a Frio** (*Cold Standby Sparing*)

Por meio da técnica de módulo redundante de espera a frio, mostrada na Figura 2.15, os módulos reserva se encontram desenergizados até o instante da comutação de um deles pelo módulo ativo em falha [70][6][131]. Quando ocorre uma falha do módulo ativo, este é desenergizado, e um dos módulos em espera é energizado e inicializado quando colocado em serviço ativo. Nesta técnica, os módulos inativos que se encontram desenergizados, por hipótese, não falham, enquanto que o módulo ativo possui uma taxa de falha constante λ .

- **Técnica de Módulo Reserva de Espera Morna** (*Warm Standby Sparing*)

Por meio da técnica de módulo reserva de espera morna os módulos reservas se encontram desenergizados até o instante da comutação de um deles pelo módulo ativo em falha [70][6][131]. A diferença em relação a técnica anterior, é que nesse caso o módulo ativo possui uma taxa de falhas λ e os módulos em espera uma taxa de falha constante φ , embora os módulos estejam desenergizados. Considera-se que $0 \leq \varphi \leq \lambda$ e que os módulos inativos quando em atividade passem a ter uma taxa de falhas λ . Quando ocorre uma falha do módulo ativo, este é desenergizado e um dos módulos em espera é energizado e inicializado quando colocado em serviço ativo. Este esquema tolerante a falhas é mostrado na Figura 2.15.

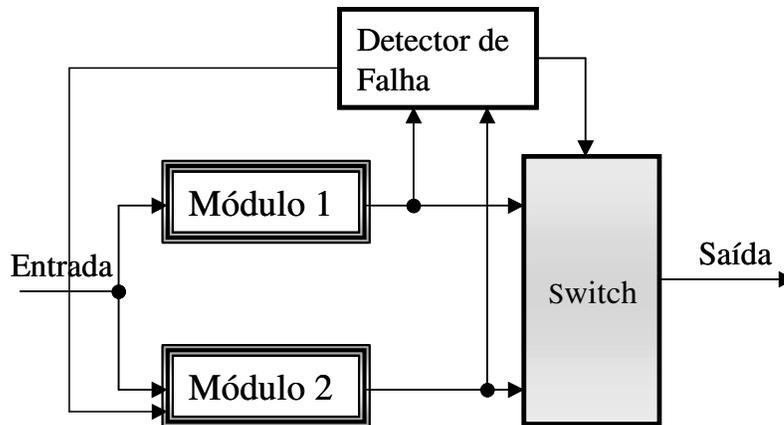


Figura 2.15 Módulo Reserva em Espera a Frio ou Morna

- **Técnicas de Redundância Híbrida de Hardware**

As técnicas de redundância híbrida combinam características essenciais às técnicas de redundância estática e dinâmica. Devido ao seu alto custo de implementação, pela quantidade de hardware envolvido, esta técnica se adequa às aplicações que necessitam uma alta dependabilidade.

- **NMR com módulos redundantes em espera (NMR *with spares*)**

Esta técnica, representada na Figura 2.16, combina as técnicas de redundância ativa e passiva, numa solução única. Ela é caracterizada pela técnica de redundância passiva NMR, composta por N módulos, com a adição de um circuito detector responsável pelo processamento da saída de todos os módulos participantes, de um circuito comutador e de um conjunto de S módulos redundantes em espera [70][77]. As saídas resultantes do processamento dos valores de entrada dos N módulos que compõem o núcleo ativo serão submetidas ao circuito votador, o qual por voto de maioria determinará qual o valor de saída final desse sistema híbrido. O detector verificará a saída de cada um dos N módulos e os comparará com a saída do votador. Os módulos cujas saídas sejam discordantes da saída do votador, serão desativados e comutados por módulos reserva, dentre os S módulos reserva que se encontram na espera. O circuito permanecerá operacional enquanto $(N+1)/2$ módulos do núcleo ativo permanecerem ativos. Caso os S módulos reserva tenham sido usados, a técnica resultante se resumirá a técnica NMR.

- **TMR/Simplex**

O desempenho da técnica TMR poderá ser incrementado, associando-se a este circuito um circuito detector, capaz de identificar o módulo falho e de desabilitar não apenas este módulo falho, mas um dos dois módulos remanescentes em operação. Portanto, após a falha de um dos três módulos de um arranjo TMR, o sistema reduzir-se-á a um esquema com dois módulos em série, sob o ponto de vista de confiabilidade. Isto acarretará numa degradação da confiabilidade. Para evitar esta situação, o circuito

detector verifica a existência de um módulo falho, descarta-o, e reconfigura o sistema para apenas um dos módulos não falhos.

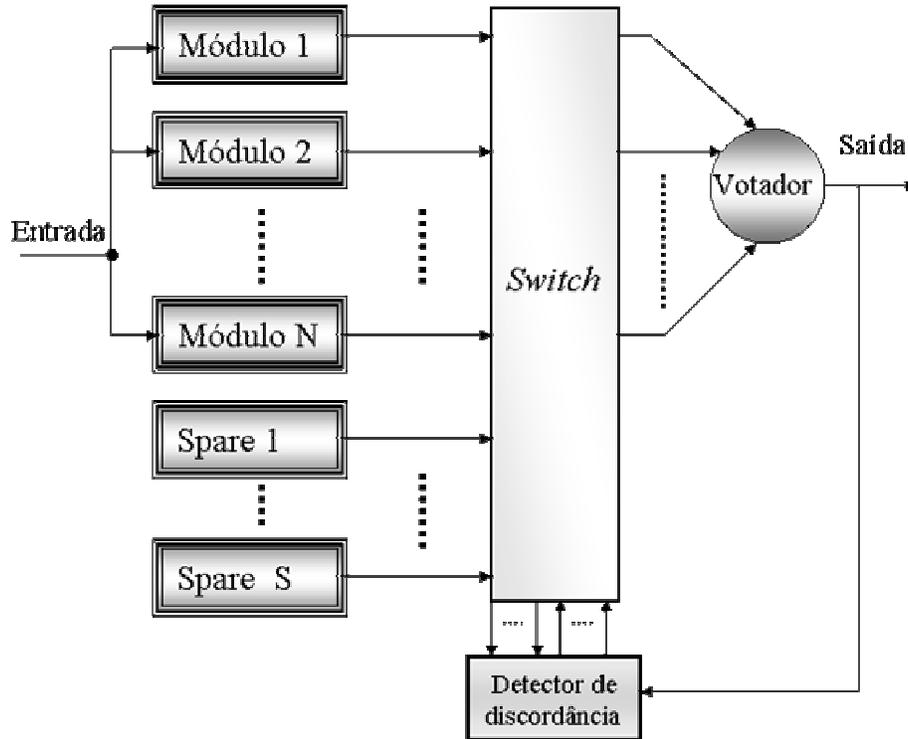


Figura 2.16 Técnica híbrida – NMR com Espera

2.6 Técnicas de Redundância de Software

Redundância de software é a adição de código extra, por meio de linhas extras de código ou pequenas rotinas de programa, ao programa original. Os softwares redundantes serão inócuos caso o sistema não venha a falhar. Redundância em software poderá ser utilizada para detecção de erros na funcionalidade do sistema por meio de testes de consistência, de capacidade, de temporização, e de replicação [6][70]. Estratégias de tolerância a falhas em software foram criadas para lidar com falhas de projeto, especialmente para os sistemas que exigem requisitos de alta confiabilidade. Entre as diversas técnicas de redundância em software, as duas principais serão descritas a seguir. Para maiores detalhes a respeito das técnicas e da implementação da tolerância a falhas em software, [115][68] poderão ser consultados.

2.6.1 Redundância Estática ou Passiva : Programação de N-Versões (NVP)

A técnica de redundância em software, denominada Programação de N-Versões (*N-Version Programming*) [32][13], é uma extensão da estrutura NMR, a qual tem demonstrado sua eficiência no estudo de tolerância a falhas em hardware. Na estrutura de hardware NMR os componentes redundantes são similares, enquanto numa estrutura

NVP as N versões de um programa, projetadas independentemente e executadas simultaneamente, devem satisfazer a uma especificação comum. As N versões são ditas independentes pois são produzidas utilizando-se diferentes algoritmos, desenvolvidas por profissionais de diferentes perfis profissionais e usando diferentes linguagens de programação, compiladores e sistemas operacionais, entre outros recursos diversos. Os resultados produzidos pela execução concorrente dos $N \geq 2$ programas, funcionalmente equivalentes, são comparados por um mecanismo de decisão, um teste de replicação que considera um resultado como válido quando a maioria das saídas forem suficientemente similares. O mecanismo de decisão se torna cada vez mais complexo, quanto mais idênticos forem os resultados a serem validados [77]. Baseado no voto de maioria o votador pode mascarar resultados errôneos, e liberar um resultado válido, de acordo com as versões majoritárias, para o restante do sistema. Se não houver acordo da maioria das N-versões, um resultado inválido é produzido para o sistema.

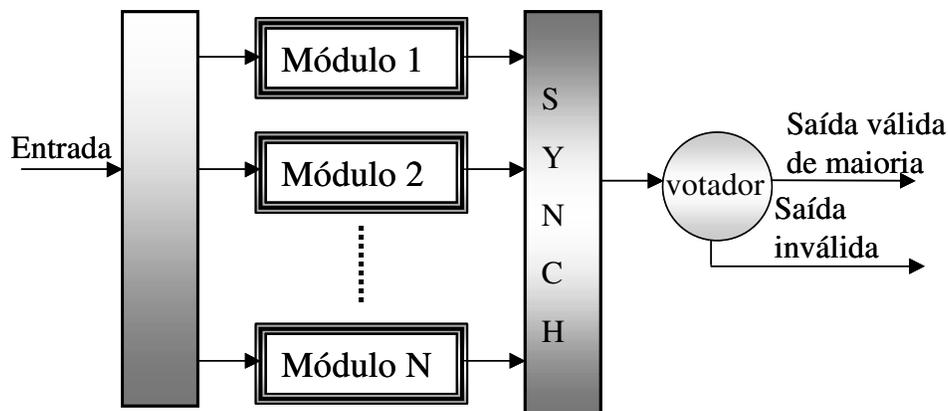


Figura 2.17 Programação a N-Versões (NVP)

O mecanismo de decisão leva em conta dois tipos de votação: o voto exato e o voto inexato. A votação exata é usada para comparações aritméticas ou entre *strings*, onde a votação majoritária de valores iguais é predominante, isto é, os resultados são idênticos bit-a-bit. A votação inexata é usada em aplicações onde a saída dos módulos da N-versões são valores numéricos, de natureza contínua. Também se utiliza a votação inexata em aplicações onde discrepâncias entre resultados são admitidas, em virtude de deficiências do hardware ou do algoritmo utilizado, desde que a diferença entre os resultados das versões estejam dentro de um intervalo de aceitação. Um programa supervisor é usado para supervisionar todas as interações entre as N versões de software e o votador, além de manusear parte do mecanismo de sincronização, o qual coloca as versões independentes em diferentes estados de operação [32][68], conforme mostrado na Figura 2.17.

Como todas as versões são geradas a partir de uma especificação inicial, esta deverá ser correta, completa e não ambígua, pois do contrário requisitos imprecisos serão gerados [50]. Portanto, todo o esforço de validação deverá ser colocado na especificação inicial, a qual não poderá ser falha, pois isso induziria os programadores a cometerem falhas de projeto semelhantes.

2.6.2 Redundância Dinâmica ou Ativa: Blocos de Recuperação (RB)

O esquema de Bloco de Recuperação (*Recovery Block*), proposto por Horning, e aperfeiçoado por Randell [68][6][118] como uma notação, permite ao programador a possibilidade de inclusão em seus projetos de software de testes próprios de aceitação, em estágios intermediários durante a execução do programa, e de mudar a seqüência de execução, caso estes testes provem ser inadequado o trecho do programa em teste. O esquema de bloco de recuperação utiliza o processo de recuperação de erro por retrocesso, o qual exige o estabelecimento de um ponto de recuperação (*recovery point*) ativo, responsável pela preservação de informações apropriadas, para posterior recuperação, em caso de erro. Os pontos de recuperação são ditos ativos a partir do momento em que forem estabelecidos até o momento em que forem descartados. O esquema de bloco de recuperação por meio do mecanismo de recuperação por retrocesso permite não apenas o reparo de resultados inválidos, como também de funções inválidas. A técnica de recuperação por retrocesso apesar de apresentar uma maior complexidade de implementação, requerendo, pois, um maior espaço em disco e um maior tempo de processamento, pode ser usada como um mecanismo de uso geral. Esta técnica de redundância dinâmica em software faz uso de rotinas alternativas independentes, funcionalmente equivalentes, para provisão de uma maior dependabilidade. Este tipo de recuperação desfaz as ações executadas e retorna a um estado consistente (*checkpoint*) no qual já tenha passado, ou pelo qual poderia ter passado, de modo a prosseguir a partir daquele ponto. Uma das grandes virtudes dessa técnica é possibilitar ao projetista a recuperação do sistema das condições de erro, sem que haja necessidade de se fazer qualquer hipótese acerca das falhas, as quais, em software, são em sua maior parte de natureza não-antecipada. Uma das desvantagens desta técnica está relacionada ao desempenho e a utilização de recursos do sistema para armazenamento das informações necessárias a recuperação.

Através do emprego da diversidade de projeto, a partir de uma mesma especificação, um módulo primário e diferentes módulos alternativos são desenvolvidos. Os módulos alternativos poderão ser escolhidos de versões prévias do módulo primário. Contudo, a independência entre os diversos módulos é um fator importante para que sejam evitadas falhas de origem comum entre eles. O bloco de recuperação é um programa composto por pontos de verificação (*checkpoints*), testes de aceitação (*acceptance tests*), e procedimentos, principal e alternativos, para uma dada tarefa. A sua estrutura, do ponto de vista da linguagem de programação ADA [118], pode ser vista conforme a Figura 2.18.

```
ensure <teste de aceitação>  
by <alternativa primária>  
else by <primeira alternativa>  
else by <segunda alternativa>  
  
else by <nésima alternativa>  
else error
```

Figura 2.18 Estrutura do mecanismo de Bloco de Recuperação.

Na Figura 2.18, observa-se que para cada bloco crítico de programa podem ser definidas uma ou mais alternativas, sendo a alternativa primária a alternativa principal a ser considerada, e as demais alternativas, se existirem, as alternativas secundárias (primeira alternativa, segunda alternativa, ..., n-ésima alternativa) [78][6]. A verificação da validade de cada uma das alternativas será realizada por meio de um teste de aceitação, o qual poderá ser único para todas as alternativas, ou específico para cada uma delas. Os testes de aceitação não garantem a correção, porém a validação dos resultados produzidos por um módulo. A estrutura dos blocos de recuperação poderá ser aninhada, ou seja, poderá incorporar outras estruturas de blocos de recuperação recursivamente [50].

O passo inicial numa estrutura de bloco de recuperação [68][6] consiste no estabelecimento de um ponto de recuperação, seguido pelo armazenamento das partes relevantes do estado do sistema nesse ponto em uma estrutura conhecida como cachê de recuperação (*recovery cache*), conforme observado na Figura 2.19, de modo a permitir a recuperação de erro por retrocesso. Após o armazenamento dos dados de recuperação no ponto de recuperação (*checkpoint*), o módulo primário é executado e os seus resultados são submetidos a um teste de consistência, denominado de teste de aceitação. Caso o teste de aceitação valide os resultados, o bloco de recuperação terminará, e os resultados serão passados como a saída do bloco de recuperação. Caso contrário, o estado inicial do sistema no ponto de recuperação, salvo na cachê de recuperação, será restaurado e um módulo alternativo secundário deverá ser executado. Esta seqüência de eventos deverá prosseguir até a produção de resultados válidos pelo teste de aceitação e liberado como saída do bloco de recuperação, ou até a exaustão de todos os módulos alternativos, quando então um sinal de erro será sinalizado para o ambiente da estrutura do bloco de recuperação. A alternativa primária é aquela através da qual se pretende realizar a operação desejada. As alternativas secundárias, normalmente, são aquelas de um custo menor e de implementação mais simples. O sistema executará uma próxima alternativa, caso esta exista, quando da não aceitação da alternativa anterior, devido, por exemplo, a uma divisão por zero, pelo término de uma temporização, ou então pela não validação dos resultados por meio do teste de aceitação. Caso uma determinada alternativa seja válida, as demais terão seus processamentos ignorados na presente execução do sistema; Caso todas as alternativas aplicadas venham a falhar, conforme dito anteriormente, um sinal de erro será enviado ao sistema. Entretanto, não haverá necessidade de se realizar um diagnóstico automático do erro, quando da ocorrência de uma falha, uma vez que o estado atual do sistema será desconsiderado, pelo retorno as condições de estado do sistema anterior à entrada no bloco, apagando, por conseguinte, todos os efeitos da alternativa defeituosa.

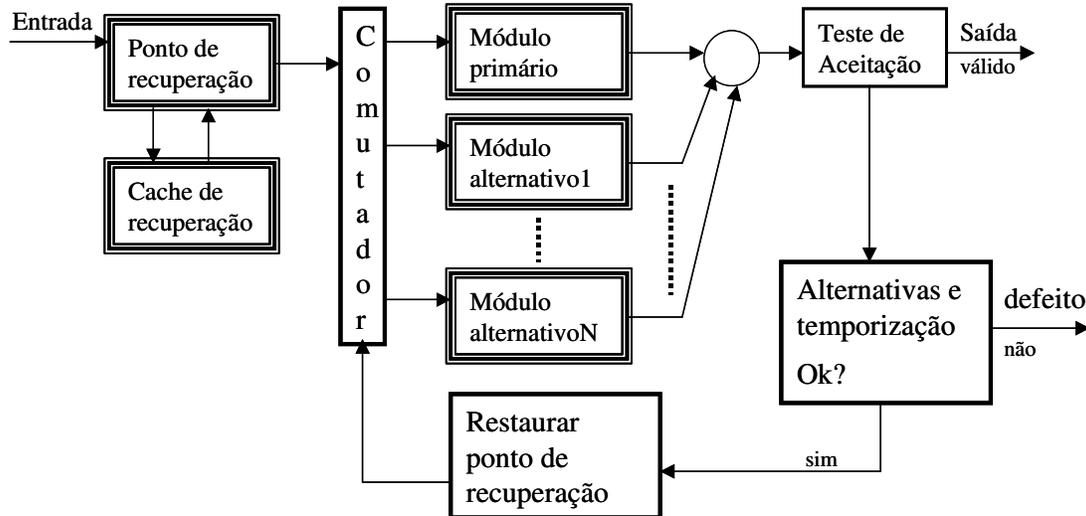


Figura 2.19 Redundância Dinâmica por Bloco de Recuperação

É importante lembrar, que os efeitos de uma alternativa anterior, não terão qualquer influência nas alternativas posteriores, em virtude de todas elas partirem de um mesmo estado inicial e, portanto, anularem os seus efeitos sobre as demais [6]. Uma das vantagens apresentadas pela técnica de bloco de recuperação, diz respeito ao aproveitamento das versões anteriores de um determinado software, as quais apresentam uma alta dependabilidade em virtude do processo de depuração a que foram submetidas durante um determinado tempo de uso. Neste caso, a nova versão incorporará a versão anterior acrescido de novas facilidades, as quais por serem novas poderão apresentar alguma falha de especificação ou de projeto. O algoritmo de recuperação a se programar faz uso de uma alternativa primária e de uma ou mais alternativas secundárias, a depender do grau de confiabilidade exigido e dos custos devido provenientes dos excessos (*overheads*) do tempo de processamento e do espaço de armazenamento dos dados na memória. Os testes de aceitação são verificações de última instância, realizado na saída do bloco de recuperação. Não é relevante saber qual das alternativas gera os resultados para o programa, contudo é necessário que estes estejam dentro do limite de aceitação. Os testes de aceitação devem ser mais simples do que as alternativas que se propõe a testar, embora não raras vezes, sejam de maior complexidade. Caso o teste de aceitação não seja projetado convenientemente, dois tipos de falhas poderão ocorrer: rejeição de alternativa correta ou aceitação de alternativa incorreta [35]. Os testes de aceitação fornecem condições para que o programador possa incorporar seus próprios mecanismos de validação para condutas errôneas do programa, adicionalmente aos testes que são intrinsecamente realizados pelo sistema.

Considerações Finais

Neste capítulo foram abordados conceitos e definições fundamentais ao entendimento dos capítulos que se seguem. Inicialmente, e de um modo conciso, foram descritos os conceitos de sistema e de modelo, além dos diversos tipos de sistemas existentes. Algumas definições de sistemas levaram em conta a dependência das variáveis

de saída em função das variáveis de entrada, da variável tempo, ou ainda da previsibilidade de suas ocorrências. Contudo, em várias ocasiões, os serviços providos pelos sistemas, podem apresentar uma natureza estocástica, razão pela qual foram definidos os conceitos de variáveis aleatórias e processos estocásticos. Os processos Markovianos, um tipo especial de processo estocástico, que leva apenas em conta o estado presente na definição do estado futuro, também foram definidos, considerando-se um espaço de estados discretos e um parâmetro de tempo contínuo, nas denominadas cadeias de Markov de tempo contínuo (CTMC). A representação dos sistemas por meio de cadeias de Markov, não raramente, pode ser uma tarefa complexa e trabalhosa. Deste modo, uma representação de alto nível, por meio de redes de Petri EDSPN, mostra-se adequada e necessária. A utilização das redes de Petri estocásticas, por meio de modelos EDSPN, para modelagem de eventos de falha e de reparo dos sistemas dependáveis, por sua vez, faz uso de mecanismos de tolerância a falhas. Logo, definições e conceitos a respeito de dependabilidade e tolerância a falhas também são necessários para o entendimento deste trabalho.

Capítulo 3

Trabalhos Relacionados

Introdução

O desenvolvimento de métodos e técnicas para avaliação de dependabilidade vem sendo proposto há décadas, em consonância com a evolução tecnológica. Os primeiros trabalhos objetivavam avaliar atributos de confiabilidade e disponibilidade, além dos aspectos de desempenho dos sistemas eletrônicos. Aspectos ligados ao desempenho e a confiabilidade do hardware foram tratados com bastante ênfase nos trabalhos iniciais [126]. A medida que os problemas associados ao hardware foram sendo melhor compreendidos e dominados, o foco direcionou-se aos aspectos ligados ao software, através de mecanismos para manipulação de falhas, em tempo de execução, por meio de diversidade de projeto [32][6][115]. Com o barateamento e a proliferação dos sistemas embarcados e computacionais de pequeno porte, do tipo PDA, telefones celulares e *smartphones* (uma combinação dos dois últimos), um novo componente dos sistemas computacionais, o ser humano, começa a ser avaliado, conjuntamente com os componentes de hardware e de software, especialmente nos aspectos relativos à sua interface com os sistemas computacionais [127]. Com a evolução contínua da complexidade dos sistemas e da infra-estrutura de interconexão entre eles, o desenvolvimento de ferramentas de avaliação de dependabilidade capazes de modelar a degradação e a reconfiguração do sistema ou da rede, de um modo dinâmico, tornam-se cada vez mais necessários. De modo a possibilitar uma avaliação mais consistente da metodologia proposta, em relação ao estado da arte e as tendências futuras das metodologias de análise e modelagem dos sistemas dependáveis, considerações a respeito de alguns trabalhos correlatos devem ser feitas.

3.1 Avaliação Determinística e Estocástica

Devido a rápida evolução das tecnologias de hardware e de software, metodologias de projeto sofisticadas são requeridas para lidar com a complexidade das arquiteturas modernas de computadores e facilitar a avaliação e a otimização dos atributos de dependabilidade [66][67][86][127]. As técnicas de avaliação de dependabilidade das arquiteturas de computadores são baseadas em modelos e medições. As técnicas de medição são constituídas por observações do sistema real em operação, também conhecida como medidas de campo, e por experimentações controladas, onde falhas são deliberadamente injetadas no sistema alvo, de forma que haja uma melhor compreensão dos efeitos das falhas sobre o sistema, usualmente referidas como experimentos por injeção de falhas [10]. Por sua vez, os modelos analíticos, dependendo do nível de abstração aplicado, podem ser caracterizados, segundo [67], como:

- **modelos determinísticos** – aplicados aos sistemas de tempo real, onde as ações ocorrem em instantes de tempo determinados ou dentro de intervalos fixos de tempo. Exemplos representativos destes modelos são as álgebras de processo temporizadas [31], as redes de Petri temporizadas [105], e os autômatos temporizados [31];
- **modelos estocásticos** – aplicados aos sistemas de compartilhamento de recursos – onde as ações são de natureza aleatória, devido, por exemplo, a ocorrência de falhas, contenciosos ou estratégias de serviço aleatória. Nesta classe de modelos estão representados, por exemplo, os modelos de fila [31][22], as redes de Petri estocástica [93][94][86], os autômatos estocásticos [31] e as álgebra de processos estocásticas [63][86].

Os modelos estocásticos [86] podem ser classificados como modelos de simulação estocástica [120][127][31] e modelos numéricos/analíticos. Estes últimos, por sua vez, podem ser divididos [86][64] em métodos combinatoriais e de forma fechada (*closed-form*), métodos de solução numérica (ou métodos de espaço de estados), e métodos numéricos aproximados, os quais incluem os métodos de decomposição hierárquica [140] e os métodos de aproximação por fases [88]. As avaliações de dependabilidade de sistemas baseadas em modelos, por meio de métodos numérico/analíticos, serão aquelas a serem primariamente referidas nesta Tese.

3.2 Métodos Baseados em Modelos Numéricos/Analíticos

De um modo geral, os métodos baseados em modelos numéricos/analíticos para avaliação de dependabilidade de sistemas podem ser divididos em dois grandes grupos [22]: a) formalismo de modelagem não baseado em espaço de estados ou modelos combinatoriais (não geram o espaço de estados); b) formalismo de modelagem baseado em espaço de estados (computam o espaço de estado dos processos estocásticos).

3.2.1 Modelos Combinatoriais (não baseados na geração do espaço de estados)

Dentre os modelos combinatoriais utilizados para modelagem e avaliação de sistemas dependáveis, podem-se considerar as seguintes soluções [89]:

- **diagrama de bloco de confiabilidade** - *Reliability Block Diagram* (RBD): é uma estrutura gráfica com dois tipos de nós: blocos, representando os componentes/subsistemas de um sistema, e nós *dummy* para conexão entre os componentes. Esta solução mapeia a dependência operacional de um sistema com respeito aos componentes/subsistemas que o compõe. Neste tipo de modelo, as falhas são consideradas independentes, de maneira que a falha de um componente não influi na falha do outro [101]. Modelos seriais e paralelos são bastante úteis não apenas por serem intuitivos, mas por terem uma dependência linear com o tempo [122]. Os modelos *m/n* [17] são uma generalização do modelo paralelo. Os modelos combinatoriais, por meio de diagramas de blocos de confiabilidade, são bastante

utilizados para modelagem de confiabilidade e disponibilidade [140][70][17], e pacotes de software, tal qual o SHARPE [122], estão disponíveis para construção e solução desses modelos;

- **árvore de falha** - *Fault Tree* (FT) [17][75][125]: é um grafo acíclico composto por nós internos, representados por portas lógicas dos tipos *AND*, *OR*, *m/n* [17], e nós externos, representados por eventos básicos que correspondem aos componentes do sistema. Os arcos representam entidades booleanas Verdadeiro e Falso (0 e 1) [106]. Quando há falha de um componente uma saída Verdade é transmitida, caso contrário, uma saída Falsa é liberada. Assim como no caso RBD, os caminhos representam a dependência operacional do sistema sobre os componentes. Em qualquer instante de tempo um valor lógico no nó raiz determina se o sistema está ou não em estado operacional. Quando a árvore de falha não possui nós compartilhados, o seu modelo equivale ao modelo serial/paralelo dos diagrama de blocos de confiabilidade (RDB) [89], porém quando o compartilhamento de nós ocorre, isto é, quando eventos repetidos são permitidos, as árvores de falha tornam-se mais poderosas [89]. Extensões de árvores de falha para solução de problemas específicos de sistema foram desenvolvidas pela introdução de portas *AND* com prioridade, portas *Exclusive-OR* e portas inibidoras [46]. Pode-se utilizar árvores de falha na modelagem de confiabilidade e disponibilidade [141], na modelagem de segurança [85] e na modelagem de tolerância a falhas em software [47]. Diversas soluções para árvores de falha com eventos repetidos podem ser encontradas em [117]. Assim como para os modelos RBD, pacotes de software foram desenvolvidos para dar suporte a construção e avaliação de sistemas dependáveis através de árvores de falha, como por exemplo, através da ferramenta SHARPE[122].
- **grafo de confiabilidade** - *Reliability Graph* (RG): tem sido bastante utilizado para modelagem de confiabilidade de redes. Um grafo de confiabilidade é um grafo acíclico representado por um conjunto de nós e um conjunto de arcos. A Falha de um componente é indicada pela eliminação do arco correspondente no grafo [101]. O sistema modelado pelo grafo de confiabilidade é considerado operacional quando houver pelo menos um caminho entre os nós fonte e dreno, os quais são nós especiais no modelo. Através do pacote de software SHARPE [122] é possível a sua construção e a análise de sistemas dependáveis.

3.2.2 Modelos baseados na geração de espaço de estados

Os modelos combinatoriais apresentam como pontos fortes uma grande simplicidade e intuitividade na modelagem de sistemas dependáveis, porém apresentam como desvantagens, a dificuldade de modelar dependência estocástica, como o processo de reparo, e a cobertura de falha imperfeita [17]. Para lidar com estas dificuldades, os formalismos de modelagem baseado em espaço de estados são mais adequados [22]. As cadeias de Markov são os modelos de espaço de estados mais comumente utilizados para modelar a dependabilidade de sistemas, além de permitir a avaliação de várias métricas relativas a dependabilidade [10]. As cadeias de Markov, apesar de serem capazes de

capturar vários tipos de dependências que ocorrem nos modelos de confiabilidade e de dependabilidade, têm como desvantagem problemas relacionados à geração do espaço de estados e a representação dos eventos não-Markovianos. Uma terceira característica indesejável dos modelos de dependabilidade Markovianos, denominada *stiffness*, pode ser reduzido pela separação do modelo de disponibilidade do modelo de desempenho na análise dos sistemas dependáveis, reduzindo com isto o problema causado pela diferença das ordens de grandeza dos tempos de falha, de processamento e de recuperação das falhas. Uma maneira de se tratar o problema de *stiffness* é denominada técnica de agregação, conforme descrito em [20], a qual faz parte de um conjunto de métodos conhecidos como *stiffness-avoidance* [101].

3.2.2.1 Modelos não-Markovianos

Modelos Markovianos podem ser adotados como uma ferramenta de modelagem bastante poderosa [136] para análise de dependabilidade, uma vez que eles são capazes de capturar várias dependências existentes nos sistemas reais. Os modelos Markovianos de parâmetro discreto (DTMC) e tempo contínuo (CTMC), por serem, em geral, matematicamente mais tratáveis do que os modelos que não têm a propriedade de ausência de memória, são mais utilizados. Em vários fenômenos a hipótese exponencial não é adequada sendo, portanto, necessária a utilização de modelos não-Markovianos [88]. Diversas técnicas têm sido propostas para representar e avaliar atividades representadas por variáveis aleatórias não-Markovianas através de modelos Markovianos. De forma geral, recorre-se a um conjunto de distribuições denominadas poli-exponenciais [59]. Os métodos de aproximação por fase [88][130], obtidos por meio de restrições impostas à classe de distribuições mais gerais definidas por Cox [42][88], ou os que adotam as distribuições hiper-exponenciais e variantes de distribuições gama, são exemplos do uso deste tratamento [140][43]. Uma outra alternativa é a utilização de variáveis complementares (suplementares) como mecanismo de registro da memória em cada estado [60], ou ainda por modelos de Markov regenerativos [87][88]. Modelos Markovianos e modelos Markovianos regenerativos têm sido bastante utilizados para análise de dependabilidade de sistemas de hardware [140], análise da confiabilidade conjunta de hardware e software [82], e na análise transiente de dependabilidade de sistemas de missão por fases [104]. Muitos pacotes de software dão suporte à avaliação destes modelos, conforme descritos [134], por meio de simulação ou análise de modelos através de aproximações por fase [145], SHARPE [121][138], que utiliza aproximações por fases, ou ainda, através de ambientes de software que suportam múltiplos formalismos de modelagem, como o Möbius [44]. É importante ressaltar ainda as soluções que evitam a explosão de estados (*largness avoidance*) e aquelas capazes de tolerar a explosão de estados (*largness tolerance*) [22].

3.2.2.2 Técnicas de *Largeness Avoidance*

As técnicas de *largeness avoidance* são usadas para evitar a explosão de estados. Técnicas do tipo truncamento de estados (*state truncation methods*) [99], solução hierárquica (*hierarchical model solution*) [79] e *decomposição* [135] são exemplos de

técnicas utilizadas para se evitar a explosão de estados. A técnica de *state lumping* [27] é bastante utilizada para redução do espaço de estados, pela exploração de simetrias estruturais nas cadeias de Markov de tempo contínuo (CTMC). O processo de *lumping* reduz o tamanho de uma CTMC pela troca de um conjunto de estados por um único estado, o qual preserva a propriedade Markoviana e possibilita a obtenção de medidas. Métodos de agregação/desagregação para análise de estado permanente ou transitório de uma CTMC podem ser vistos em [22]. Estes métodos podem ser usados para eliminar propriedades indesejáveis do modelo, tais como *stiffness*, como foi dito anteriormente. Técnicas denominadas *importance sampling* são usadas para acelerar artificialmente eventos raros de uma maneira controlada, fazendo algumas medidas corretivas posteriores [107][22]. Em [19] é apresentado um método de análise numérica, por software, para pequenos modelos, e uma simulação rápida por meio de *importance sampling*, para grandes modelos, através de CTMC gerada pela ferramenta de modelagem de alto nível, SAVE [62]. Nesta ferramenta o sistema é considerado uma coleção de componentes, os quais podem ser dispositivos de hardware, software, estruturas de dados ou componentes ambientais (unidade de resfriamento, de potência, etc...).

3.2.2.3 Técnicas de *Largeness Tolerance*

As técnicas de *largeness tolerance* são utilizadas para dar suporte de modelagem para as CTMCs que geram grande quantidade de estados, através de um modelo conciso em um alto nível de abstração. Em geral, estas técnicas são baseadas em extensões de redes de Petri estocástica [86] ou álgebra de processo estocástica [63], pelo desenvolvimento de algoritmos capazes de manipular CTMC com grande quantidade de estados, por meio de estruturas de dados, por exemplo, do tipo *Bynary Decision Diagram* (BDD) [25], associadas a modelagem composicional. Caso sejam assumidas algumas hipóteses, como por exemplo que os modelos individuais, isoladamente, terão o mesmo número de estados quando em conjunto com os demais sub-modelos, além de restrições estruturais, é originada a propriedade lógica da forma produto [39]. Uma outra solução, baseia-se na estratégia de dividir para conquistar, onde a matriz que descreve a CTMC é representada por um conjunto de matrizes menores que são apropriadamente combinadas. Estas técnicas são conhecidas como representação de Kronecker, desde que satisfeitas certas restrições estruturais [28]. A técnica *on-the-fly* [22] evita o armazenamento da matriz geradora pela regeneração dos elementos da matriz quando necessários em um algoritmo de solução iterativo.

3.3 Trabalhos Centrados na Visão do Usuário

Diante das novas tendências dos sistemas dependáveis, nas quais o usuário é uma das partes fundamentais, o trabalho de [65] mostra-se adequado. Com ênfase na visão do usuário, observa-se o impacto da recuperação de falhas nas medidas de confiabilidade, disponibilidade e performabilidade. A confiabilidade, nesse trabalho, leva em conta que o sistema não deva falhar num determinado intervalo de tempo, ou que ele se recupere voltando ao mesmo estado anterior a falha como se ele não tivesse sido interrompido. Isto

altera as medidas tradicionais de dependabilidade e desempenho, e define novas medidas sob a ótica do usuário, como *service reliability*, *service availability* e *customer performability*. A medida de *service reliability* estende a noção de confiabilidade, considerando também a possibilidade do sistema se recuperar de uma falha pelo retorno ao estado anterior a sua ocorrência. Isto é, do ponto de vista do usuário é necessário que o sistema execute a tarefa por ele solicitada, não importando se de forma contínua. A medida de *service availability* indica que o sistema deve estar não apenas operacional, mas pronto para atender ao usuário no instante de sua solicitação. A medida de *customer performability* considera a medida conjunta de desempenho e *service availability*. As medidas tradicionais utilizam modelos matematicamente tratáveis, enquanto as medidas sob o ponto de vista do usuário, apesar de apresentar maior dificuldade para avaliação analítica, podem ser avaliadas por simulação. Ainda considerando-se a visão do usuário, diferentes formalismos de modelagem têm sido propostos com o objetivo de avaliar a confiabilidade conjunta de hardware e software de sistemas computacionais, distribuídos e tolerantes a falhas. Dentre estes, o formalismo denominado modelos de ação (*action models*) descrito em [98], procura lidar com a complexidade dos sistemas e o tempo cada vez menor para lançamento no mercado, de um modo que seja rápido e de fácil tratamento por não-especialistas. As soluções dos modelos de ação está fundamentada em algoritmos baseados em caminhos. A métrica desejada está baseada na probabilidade de sucesso de execução de um *job* submetido pelo usuário ao sistema. Os modelos de ação definem, para cada *job* oferecido ao sistema, um fluxo através deste, na forma de uma seqüência de tarefas. Modelos de ação combinam aspectos de redes de Petri com o formalismo RDB, e está focada no usuário e não no sistema.

Os sistemas embarcados apresentam dificuldade na avaliação de sua dependabilidade em virtude dos requisitos de hardware e software, além da interface homem-máquina. Estes três fatores interagem de tal modo que afetam a dependabilidade de um sistema. Considera-se além da predição de confiabilidade pela composição dos componentes individuais, a possibilidade do usuário contornar uma situação de falha parcial. A avaliação de dependabilidade, na visão do usuário, pode ser definido como a habilidade do usuário interagir com um sistema e realizar com sucesso a sua missão, apesar de falhas parciais do sistema, devido a falhas dos componentes, ambientes extremos, erros de projeto de software, ou outras causas. Dentro da visão do usuário, tal definição é mais palpável que um simples número. Para tanto é sugerido em [84] a avaliação de sucesso de uma missão do usuário em termos da flexibilidade na seleção de uma série de tarefas para a obtenção da meta especificada. Com este objetivo, grafos de estados e tarefas do usuário são criados, os quais podem ser analisados por técnicas de análise de grafos.

3.4 Trabalhos utilizando formalismo estocástico, em especial, redes de Petri

A seguir serão descritos uma série de trabalhos correlatos, por ordem cronológica, que levam em conta técnicas utilizadas na modelagem e avaliação de sistemas dependáveis, em especial aquelas por meio de redes de Petri estocástica e suas extensões,

Em [5], uma metodologia para predição e avaliação de confiabilidade, manutenibilidade, e disponibilidade, é desenvolvida, através de modelos GSPN, por agregação de estados. Baseado na hipótese do tempo de reparo das atividades do sistema ser muito menor do que o tempo entre falhas, um método de decomposição por escala de tempo é implementado no modelo GSPN, o que permite a obtenção de uma representação hierárquica das atividades do sistema. Pela técnica de decomposição hierárquica o sistema é decomposto em uma seqüência de subredes agregadas, cada uma das quais sendo válida em uma certa escala de tempo. Isto ajuda a reduzir o espaço de estados e a possibilidade de *stiffness* das cadeias de Markov correspondentes. Em [70] são descritas uma série de técnicas de tolerância a falhas de sistemas, bem como modelos combinatoriais e baseados em estados (Markovianos) de confiabilidade, disponibilidade, segurança e manutenibilidade, para a avaliação da dependabilidade dos sistemas. Também é apresentado um estudo de caso que permite avaliar a melhor solução de projeto, dentre várias arquiteturas propostas, com relação a um sistema de controle eletrônico de vôo de uma aeronave. Os valores de confiabilidade das diversas arquiteturas foram obtidas através de análises dos modelos de Markov. Em [119] são discutidos métodos para predição da confiabilidade de sistemas por meio de árvores de sucesso e cadeias de Markov, através de uma estratégia de modelagem, interativa e hierárquica, na qual os modelos podem ser refinados de acordo com a necessidade.

A modelagem de um sistema de manufatura flexível real, por meio de GSPN, é proposto em [95]. O sistema é decomposto de forma a reduzir a sua complexidade e permitir que avaliações de desempenho e dependabilidade sejam realizadas. O sistema de manufatura é decomposto em sistema de produção e ambiente. Um modelo de mais alto nível, denominado modelo de dependabilidade é gerado a partir dos dados fornecidos pelos modelos de nível mais baixo associados ao desempenho e a dependabilidade. [29] apresenta um resumo de algumas ferramentas de modelagem capazes de especificar modelos de sistemas computacionais degradáveis e com habilidade para analisar atributos de desempenho e dependabilidade dos mesmos. São mostrados modelos em redes de Petri do tipo SPN e GSPN, para diferentes combinações de falha e reparo. São discutidos em [88], métodos para análise de cadeias semi-Markovianas por meio de solução numérica, simulação de evento discreto e aproximação por fases, este último, aplicado aos casos de distribuições determinísticas, weibull e lognormal. Técnicas para análise de confiabilidade e performabilidade utilizando-se os modelos MRM, QN e GSPN e ferramentas diversas como METFAC, NUMAS, SHARPE, SPNP, TANGRAM, UltraSAN entre outras, são descritas em [137]. Em [89] é estabelecida uma hierarquia formal entre os modelos de dependabilidade combinatoriais e aqueles orientados a espaço de estados (Markovianos), com relação a potencialidade de modelagem. Foram definidos algoritmos que permitem a conversão dos modelos combinatoriais (RDB, FT, RG e FTRE) entre si, bem como algoritmos que permitem a transformação recíproca entre os modelos Markovianos (CTMC, GSPN, MRM, SRN).

Um trabalho muito interessante pode ser encontrado em [135], onde são discutidas regras de modelagem e validação dos sistemas do tipo *life-critical* em tempo real, por meio de uma metodologia de modelagem, multi-nível, através de modelos DSPN.

Através desta técnica, modelos de subsistemas são construídos com a finalidade de reduzir a dimensão do espaço de estados ou o problema de *stiffness*, de uma forma separada. Os resultados obtidos são usados como parâmetros nos modelos de nível mais alto. Esta metodologia foi aplicada, de um modo eficiente a um sistema de resfriamento tolerante a falhas em tempo real de uma usina nuclear, pela divisão do sistema em três subredes, dependentes e concorrentes: subsistema de estado operacional (de mais alto nível), subsistema de inspeção e subsistema de bombeamento (de mais baixo nível). [100] estuda a avaliação de confiabilidade e de disponibilidade para situações de falha comum, taxas de falha dependente do estado do modelo para situações de mudança de carga do processador, política de manutenção e cobertura de falha imperfeita dos processadores. Pode-se observar em [90] uma metodologia para construção de modelos dependáveis utilizando-se os modelos GSPN e SRN, e também algoritmos para transformação de árvores de falha nos modelos equivalentes GSPN e SRN. Modelos de reparo são introduzidos assumindo-se a possibilidade da dependência de reparo entre os componentes. Algumas políticas de filas para reparo, tais como FCFS (primeiro a chegar, primeiro a ser servido), prioritária com preempção, prioritária sem preempção e reparo com compartilhamento de processador são consideradas para sistemas onde haja dependência de reparo, através dos modelos GSPN e SRN.

Modelagem e avaliação de dependabilidade por meio de cadeias de Markov aplicadas às técnicas de tolerância a falhas em software, considerando-se problemas de confiabilidade e segurança, podem ser vistas em [9]. As técnicas utilizadas neste estudo, bloco de recuperação e programação a n-versões, são as técnicas de tolerância a falhas em software mais utilizadas e das quais derivam uma série de outras soluções. [87] trata a modelagem dos processos Markovianos regenerativos (MRGP) por meio de modelos Markovianos regenerativos (MRRM) ou redes de Petri DSPN, e sua utilização para análise de desempenho e confiabilidade dos sistemas computacionais e de comunicação, sem a explosão de estados, comum aos modelos de aproximação por fase. Em [47], uma metodologia de modelagem de dependabilidade é apresentada, a qual define uma solução hierárquica baseada na combinação de técnicas de árvores de falha e processos de Markov, para análise de sistemas tolerantes a falhas compostos por um certo número de réplicas de hardware, um determinado número de variantes de software e um algoritmo de decisão. O comportamento do sistema a longo prazo é representado por um modelo Markoviano, no qual os estados da cadeia de Markov representam a evolução da configuração do hardware, a medida que falhas permanentes ocorrem e são tratadas. Ou seja, cada estado da cadeia de Markov representa uma configuração particular dos componentes de hardware e de software e, portanto, um diferente nível de redundância. O comportamento do sistema a curto prazo é representado por modelos de árvore de falhas, os quais capturam as falhas de software e as falhas transientes de hardware no processo de computação. Por meio desta metodologia de modelagem, análise quantitativa de confiabilidade e de segurança são realizadas para diferentes arquiteturas de sistema tolerante a falhas. No modelo de confiabilidade, qualquer resultado inaceitável é considerado uma falha. No modelo de segurança, assume-se que um erro detectado é tratado pelo sistema em um modo de falha segura, e que um resultado inseguro ocorre somente se um resultado inaceitável não for detectado.

Métodos para análise de dependabilidade de sistema são mostrados, métodos combinatoriais dos tipos RDB, FT e RG são discutidos e técnicas de geração de espaço de estados, representadas pelos modelos CTMC, GSPN e SRN, são descritos em [139]. Também modelos não-Markovianos são apresentados pela utilização de variáveis suplementares, expansão por fase e teoria da renovação de Markov (seqüência de renovação, processos semi-Markovianos e processos regenerativos). Em [112] é discutida a sensibilidade das estimativas de dependabilidade em relação ao fator de cobertura e de sua distribuição com relação ao tempo quando da ativação dos mecanismos de detecção de erro. São propostos modelos exponencial e não-exponencial para a latência da cobertura.

Em [72][74] é apresentada a modelagem de dependabilidade dos sistemas tolerantes a falhas constituídos por hardware e software, levando-se em consideração as interdependências entre os componentes. Este método consiste na construção modular e sistemática de modelos tolerantes a falhas utilizando redes GSPN, pela composição de sub-modelos de seus componentes de hardware e software e suas interações. As principais interações entre os componentes de hardware e software são identificadas com o objetivo da construção de um modelo de blocos do sistema. Este modelo é constituído pelos blocos dos componentes e pelos blocos de dependências entre estes. Em seguida estes blocos são transformados nas GSPNs correspondentes aos componentes e suas dependências. Regras de composição são definidas as quais permitem flexibilidade e reusabilidade dos modelos GSPN. [110] apresenta a avaliação de confiabilidade e performabilidade em tempo real, aplicada à técnica de tolerância a falhas em software do tipo programação a N-versões (NVP), por meio de decomposição hierárquica, considerando-se um determinado período de missão. Avaliação de dependabilidade por meio de redes GSPN aplicada a sistema de controle computacional de tráfego aéreo pode ser vista em [56]. A solução de modelagem consiste na modelagem do sistema como um conjunto de módulos interconnectados por meio de mecanismos de acoplamento. Nesta solução, o modelo é construído e validado de um modo incremental, de acordo com um conjunto de diretrizes. A cada passo, a metodologia de modelagem incorpora um novo componente, atualiza e valida o modelo GSPN correspondente.

Em [102], modelos em redes de Petri determinística e estocástica (DSPN) são utilizados para modelagem e avaliação de dependabilidade em sistemas de missão por fases (PMS). Devido a sua natureza dinâmica, os sistemas PMS variam de acordo com o seu comportamento de uma fase a outra, e ainda devido as condições do ambiente nos quais estão envoltos. O uso das DSPNs possibilita a avaliação dos modelos com atividades exponencialmente distribuídas, determinísticas e transições imediatas. O modelo é conciso e de fácil entendimento. As modificações e reconfigurações dos sistemas PMS, por meio dos modelos DSPN, são descritas em função dos predicados dependentes da marcação, o que proporciona a estes sistemas flexibilidade devido a facilidade de adaptação. Uma metodologia de modelagem e avaliação, hierárquica e modular, para sistemas de missão por fases é apresentada, a qual é baseada em processos de Markov, pode ser vista em [103]. Nesta metodologia as fases têm uma duração constante e pré-determinada. O processo de modelagem é dividido em dois níveis: o nível mais alto modela a missão propriamente dita, enquanto o nível mais baixo modela as

várias fases. A modelagem e a resolução das fases são consideradas de forma separada, das dependências entre elas. As vantagens dessa solução estão na flexibilidade, facilidades de aplicação, e reusabilidade dos modelos definidos. No nível de modelagem mais alto, um modelo único é construído para toda a missão, caracterizado por um conjunto de fases, sem que haja detalhamento do comportamento do sistema dentro de cada fase. Os valores dos parâmetros a serem usados no nível mais alto são provenientes do nível mais baixo. Os modelos de nível mais baixo detalham o comportamento do sistema nas diversas fases, e são construídos e resolvidos separadamente. Portanto se uma fase é repetida durante uma missão, o modelo correspondente pode ser construído uma única vez e reusado quando necessário. O comportamento do sistema, tal como visto nas fases, é caracterizado pela execução de tarefas, as quais podem ter altos ou baixos requisitos de dependabilidade, dependendo do nível de criticidade das operações envolvidas. Os mesmos componentes, presentes em uma fase, podem ter diferentes intensidades de falha e reparo em outras fases. Contudo, numa mesma fase as taxas de falha e de reparo dos componentes, assim como os requisitos do sistema, permanecem constantes. O modelo de nível superior é uma DTMC, enquanto os modelos de nível inferior são construídos por meio de GSPN. As duas soluções são combinadas por meio da modelagem hierárquica. A solução da DTMC do nível superior calcula as medidas de dependabilidade de interesse. Em [104] é proposta a análise transitória de dependabilidade dos sistemas de missão por fases (PMS) por meio de modelos. Estes modelos, em função das mudanças de configurações nas diversas fases, apresentam problemas desafiantes na avaliação analítica de desempenho e dependabilidade. As taxas de falhas são dependentes das características dos ambientes em que as fases se encontram em um determinado período. Para lidar com esta situação utiliza-se uma metodologia de modelagem baseada nos modelos de rede de Petri estocástica Markovianas regenerativas. A técnica de solução analítica obtida apresenta baixa complexidade computacional e permite a dedução de funções de sensibilidade para estimativas de dependabilidade de tais sistemas.

Uma metodologia de modelagem multicamadas, baseada em decomposição hierárquica modular, utilizando diferentes métodos e ferramentas de modelagem nos diversos níveis de hierarquia pode ser visto em [23]. Através da metodologia de modelagem, somente os detalhes relevantes são considerados, como forma de se reduzir o número de estados gerados e de se evitar problemas de explosão de estados. Portanto, a metodologia começa com modelos simples, os quais se tornam cada vez mais complexos, pelo relaxamento de algumas hipóteses que causam pouco impacto nos resultados obtidos. A estruturação da metodologia por meio de diferentes níveis e separados por interfaces identificadas permite prover cada nível com metodologias e ferramentas diferentes. Cada nível, por sua vez, pode ser sub-dividido em sub-níveis para uma análise mais detalhada de algumas características. A estratégia de modelagem é baseada em:

- diferentes métodos e ferramentas que podem ser utilizados para modelagem em diferentes níveis da hierarquia, de acordo com a conveniência;
- cada modelo, de forma a reduzir a complexidade, tem pequena dimensão computacional;

- aspectos específicos estão confinados em uns poucos sub-modelos, e não requerem a redefinição do modelo por completo.

A aplicação desta metodologia ao modelo do núcleo de segurança de um sistema de controle de sinalização de uma estação ferroviária divide o modelo em duas partes: a primeira parte trabalha com uma única execução e calcula as probabilidades de sucesso e de falha; a segunda parte, de posse dessas probabilidades da parte anterior, avalia os atributos de dependabilidade para um missão inteira. Dois modelos foram adotados na construção dos modelos de uma única execução: DTMC e SAN. Diversas avaliações podem ser realizadas: a) verificar se o sistema satisfaz seus requisitos; b) analisar a sensibilidade em função de vários parâmetros; c) avaliar o impacto das várias hipóteses nas características de dependabilidade.

Em [79] são descritos modelos de análise de dependabilidade baseados em estados por meio de composição hierárquica e agregação. Esta solução permite a análise de disponibilidade e performabilidade de sistemas complexos, como os sistemas de telecomunicações, através do particionamento de modelos complexos em uma hierarquia de sub-modelos. A solução descrita explora a natureza hierárquica desses sistemas de telecomunicações, e a independência de seus subsistemas. O método descrito em [116] propõe a modelagem e a avaliação de atributos de dependabilidade para um sistema multiprocessador de propósitos gerais, pela separação explícita entre os aspectos de arquitetura, daqueles relativos ao ambiente de serviço a ser suportado. O modelo arquitetural é composto pelo modelo que descreve o comportamento dos componentes de hardware e software do sistema, bem como dos mecanismos de tolerância a falhas, e pela camada correspondente aos modos de falha do sistema; o modelo ambiental é composto pelo modelo do nível de serviço e pelo modelo correspondente a política de manutenção. A separação do modelo arquitetural do modelo ambiental permite a reutilização do mesmo modelo de arquitetura para diferentes modelos de ambiente, bem como a reutilização do mesmo modelo ambiental para diferentes modelos de arquitetura. Por outro lado, a construção modular do modelo de componentes permite o reuso de alguns sub-modelos de componentes no modelo do componente. A interconexão entre modelos arquitetural e ambiental é feita pela camada correspondente aos modos de falha, a qual além dos modos de falha, contém lugares com informações necessárias aos modelos ambientais ou dos modelos ambientais. Entretanto, a utilização dos lugares que representam informações do modelo ambiental ou arquitetural pode provocar uma explosão combinatorial do espaço de estados da cadeia de Markov em níveis mais baixo, o que dificulta o tratamento dos modelos.

Em [17] são descritos vários modelos probabilísticos para avaliação quantitativa de dependabilidade. Por meio das técnicas de modelagem combinatoriais, representadas pelos modelos RBD e FT, equações analíticas para estimativas de confiabilidade e disponibilidade do sistema são mostradas em função dos seus componentes. Técnicas de modelagem por enumeração de estados também são citadas e expressões analíticas para avaliação de dependabilidade são descritas por meio das CTMC, de nível de abstração mais baixo, e por formalismos de um nível de abstração mais elevado, representado pelas redes estocásticas bem formadas, ou *Stochastic Well formed Nets* (SWN). Os modelos em

SWN para dependabilidade seguem uma sistemática composicional para sua construção, utilizando-se de diagramas de classe UML.

A solução de análise e modelagem proposta em [1], diferentemente das demais, aplica-se à avaliação de sistemas dependáveis cuja arquitetura não seja conhecida a priori. O método de modelagem tem como restrições a indisponibilidade da maior parte das informações para cada uma das arquiteturas candidatas, e níveis desiguais de informação para cada uma delas. O método descrito em [1], denominado modelagem descendente, permite uma análise comparativa das possíveis soluções e um refinamento da arquitetura de sistema selecionada. O método segue uma solução hierárquica multi-fases, a qual parte de um modelo funcional utilizado na construção do modelo de dependabilidade de alto-nível, associado às várias arquiteturas candidatas, utilizando regras formais de construção bem definidas. O modelo de dependabilidade de alto nível é constituído por dois níveis: a) nível funcional que descreve os modos de funcionamento/degradação do sistema, bem como as dependências entre as diferentes funções; b) nível estrutural que descreve o comportamento do sistema por meio da interação dos componentes de hardware e software para realização das funções do sistema. Para cada função do modelo funcional são identificados: a) os componentes de software a serem executados; b) os componentes de hardware sobre os quais os software são executados; c) os mecanismos de dependabilidade baseados nas redundâncias [1].

Baseado na avaliação de dependabilidade, uma das arquiteturas é escolhida, e o modelo correspondente de dependabilidade de alto nível é refinado, conforme os detalhes da arquitetura escolhida vão sendo obtidos [3]. Os modelos de dependabilidade de alto nível, de cada arquitetura, são processados por meio de GSPN [94], devido a habilidade desta ferramenta em lidar com modularidade e refinamento de modelos, de modo a se obter medidas de dependabilidade. Para simplificar a construção dos modelos dependáveis de alto nível das arquiteturas candidatas, e uniformizar as análises, uma biblioteca de modelos básicos é criada, a qual com poucas modificações permite a obtenção de diferentes modelos de sistemas. Esta biblioteca é composta pelos seguintes modelos: a) unidade básica (formada por um único componente de hardware e um único componente de software); b) redundância passiva; c) redundância semi-ativa; d) redundância ativa [3].

Os modelos de alto nível de dependabilidade criados são refinados progressivamente. O refinamento pode ser de três tipos: componente, evento/estado e distribuição [4]. O refinamento pode conduzir a problemas de *stiffness* o qual é minimizado, por exemplo, pela técnica de agregação de lugar definida em [5]. A precisão do resultado está condicionada à relação entre transições lentas e transições rápidas. Quanto menor a relação, melhor será a precisão dos resultados. A dependência entre os componentes é resolvida por meio de solução de modelagem hierárquica, onde o componente a ser analisado, é dividido em subconjuntos, os quais são analisados isoladamente e os resultados são combinados para obtenção das medidas de dependabilidade do sistema. O objetivo final é avaliar os atributos de dependabilidade de um modo tão preciso quanto possível e realizar estudos de sensibilidade. Este método tem como limitação a suposição de que as taxas de falha e de reparo para todas as réplicas dos modelos da biblioteca básica sejam as mesmas, o que limita a aplicabilidade dos modelos.

3.5 Novas Tendências

Segundo [129], pesquisas recentes em dependabilidade cobrem um espectro que vai dos sistemas embarcados de tempo-real às grandes redes de arquiteturas abertas. Para sistemas reais, dois grandes problemas na previsão de falhas são: a) estabelecimento de modelos mais coerentes e tratáveis com o comportamento do sistema; b) procedimentos de análise que permitam que o modelo (possivelmente muito grande) seja processado. O processo de análise deve iniciar tão cedo quanto possível, ainda na fase de projeto, para que se possa tomar decisão entre soluções alternativas. Pesquisas na área de avaliação analítica incluem: sistemas de múltiplas fases de operação, e aplicações baseadas em internet, requerendo solução de modelagem hierárquica. Na tolerância a falhas em software, quando se avalia a dependabilidade por meio de técnicas de diversificação para tolerar falhas de projeto, atenção deve ser dada a modelagem de dependências.

Dentre as tendências apontadas por [129] estão a reconfiguração dinâmica baseada em modelo de sistemas críticos complexos, por meio da construção de modelos simplificados do sistema a ser controlado, de modo a tomada de decisões apropriadas *on-line*, de forma rápida. Baseado nos valores críticos assumidos por indicadores, relacionados aos sintomas de alarme, uma ação de reconfiguração é disparada por um gerenciador de dependabilidade. Portanto, a metodologia deve ser flexível o bastante para que possa ser facilmente adaptável a diferentes problemas.

Os novos desafios para as metodologias de modelagem estão relacionados com a crescente complexidade e dinamismo dos sistemas, sob o ponto de vista do sistema e do modelo de solução. Os problemas relacionados com a explosão do espaço de estados têm suas soluções direcionadas para os métodos modulares e hierárquicos. As soluções hierárquicas, facilitam a construção de modelos, agilizam sua solução, favorecem a escalabilidade, controlam a complexidade (por não tratar em um nível mais abstrato detalhes de um nível mais refinado). Informações de um nível devem ser agregadas em um modelo abstrato de nível superior. Problemas surgem quando se procura definir a forma para se abstrair as informações relevantes de um nível para o nível superior e como compor os modelos abstratos no nível superior. A composição é favorecida pela integração de modelos menores (blocos) na formação do modelo completo em cada nível de hierarquia, desde que sejam satisfeitos requisitos específicos da aplicação. Com relação a composição, as seguintes metas são objetivadas:

- I. Utilizar diferentes modelos de blocos para diferentes tipos de componentes no sistema. Estes blocos podem ser usados como uma biblioteca de modelos;
- II. Instanciar automaticamente um determinado modelo, um para cada componente;
- III. Em um dado nível hierárquico, automaticamente ligá-los (por meio de um conjunto de regras as quais são dependentes da aplicação), assim definindo o modelo completo.

O objetivo da computação tolerante a falhas é o desenvolvimento de sistemas computacionais confiáveis que continuem a operar em níveis satisfatórios mesmo na

presença de falhas de componentes ou de subsistemas. Portanto, métodos e metodologias deverão ser reconsiderados, estudados, explorados, avaliados e aplicados para mover os sistemas a um “ambiente de dependabilidade” [83] o qual, além de critérios de dependabilidade, também leva em consideração critérios de ergonomia, usabilidade, educação, sociologia, leis, e governos.

Diversas linhas de pesquisa recentes procuram se antecipar as tendências que advirão da contínua evolução da tecnologia, propondo soluções para problemas emergentes ou futuros. A medida que as características da tecnologia microeletrônica tornam-se cada vez mais próximos dos seus limites, novas tecnologias devem ser pensadas para lidar com as perspectivas futuras da computação. Entre as tecnologias emergentes, a computação bio-inspirada e a computação quântica têm se apresentado como soluções válidas, conforme descrição em [114]. Técnicas de dependabilidade apoiadas na detecção e correção de erros são vitais na computação quântica devido ao efeito destrutivo do ambiente, o qual age como um gerador de erros. De modo a manter os erros de propagação sob controle dentro dos processos de computação quântica, processos de recuperação seguros baseados em detecção e correção de erros devem ser fornecidos.

Considerações Finais

Este capítulo mostrou uma grande variedade de métodos que são utilizados para análise de sistemas dependáveis, envolvendo atributos os mais variados, em geral voltados para avaliação de confiabilidade, disponibilidade, segurança e desempenho, além de atributos compostos, como a performabilidade. Foram referenciados trabalhos apoiados em técnicas combinatoriais, que apesar de eficientes e intuitivas, não oferecem condições para análises de dependência entre os componentes do sistema, além de uma série de trabalhos relacionados com as cadeias de Markov e suas representações de mais alto nível, descritas pelas redes de Petri estocásticas e suas extensões. Foram tratados ainda, trabalhos envolvendo um novo componente dos sistemas atuais, o qual interage cada vez mais com os componentes de hardware e de software dos sistemas, podendo causar falhas das mais diversas por inabilidade ou malevolência, o usuário. Através de alguns trabalhos nesta direção, observa-se a definição de novas métricas e de novas visões sistêmicas apoiadas no usuário. Foram apresentados também alguns problemas inerentes aos sistemas Markovianos, como a possibilidade de explosão de estados e *stiffness*, tratadas através de técnicas de intolerância ou de tolerância a estas características, além da necessidade de se expressar eventos que não se comportem de acordo com os princípios de Markov, os denominados eventos não-Markovianos. Ao final foram apresentadas novas tendências em direção aos ambientes dependáveis e novas fronteiras a serem superadas, representadas pelos sistemas tolerantes a falhas bio-inspirados, e em especial, pelos sistemas quânticos.

Capítulo 4

Metodologia de Análise e Modelagem de Dependabilidade

Introdução

A análise de dependabilidade, por meio das redes de Petri estocásticas, objetiva não apenas uma avaliação qualitativa do sistema com respeito a aspectos de ausência de *deadlock*, limitação e reversibilidade, como também uma análise quantitativa, por meio de uma descrição compacta de alto nível. As redes de Petri estocásticas permitem, por meio de um formalismo gráfico, a análise comportamental dos sistemas caracterizados por concorrência, sincronização, exclusão mútua e conflito, aspectos típicos de ambientes distribuídos e que apresentam dificuldade de serem capturados por outros métodos. Apesar de todas essas características positivas, observa-se que as redes de Petri estocásticas são abstrações de alto nível dos processos estocásticos, sejam eles Markovianos, semi-Markovianos, ou mesmo não-Markovianos. Portanto, qualquer avaliação quantitativa dos sistemas por meio de métodos analíticos ou numéricos, requer uma análise do processo estocástico de nível mais baixo que o suporta, o que poderá levar o sistema a uma explosão de estados, dificultando a obtenção de resultados.

Conforme observado em capítulos anteriores, as técnicas de avaliação de sistemas, através de métodos combinatoriais, não requerem a geração de espaço de estados. Apesar de eficientes e bastante utilizados em processos de engenharia, os métodos combinatoriais, em geral, não conseguem capturar aspectos de dependência estocástica e nem as coberturas de falhas, razões pelas quais técnicas que capturem tais dependências são necessárias. A metodologia desenvolvida procura agregar a eficiência e a intuitividade dos modelos combinatoriais, representados pelos diagramas de blocos, com as potencialidades das redes de Petri estocásticas, na análise de sistemas dependáveis. A complexidade de avaliação e modelagem desses sistemas é resolvida de maneira modular e hierárquica, a qual, além de apresentar flexibilidade e reusabilidade de seus modelos, limita a geração do espaço de estados. Por meio da metodologia desenvolvida, observa-se a transformação de um diagrama de blocos de sistema convencional, em um diagrama de sistema dependável representado por uma rede de Petri, após várias etapas de transformação. A idéia do trabalho consiste numa especificação inicial do sistema, por meio de um diagrama de blocos, devido à sua eficácia e praticidade. Em seguida, transformações estruturais conduzem a uma rede de Petri parametrizada, a qual apresenta características de tolerância a falhas, controlada por meio de parâmetros diversos. A avaliação da rede parametrizada permite a obtenção de estimativas de confiabilidade, disponibilidade e segurança do sistema como um todo, além de flexibilizar diversas configurações do modelo do sistema dependável, e de permitir a reutilização de partes do modelo do sistema, nos modelos de outros sistemas (reuso). A complexidade do sistema e

a dimensão do espaço de estados do modelo estocástico correspondente ficam reduzidas à complexidade e à dimensão do espaço de estados do modelo correspondente a cada bloco, quando se avaliam, numérica ou analiticamente, os modelos dos diversos blocos que compõem o diagrama do sistema, isoladamente, e também, na maior parte das vezes, individualmente.

Antes de serem mostradas as etapas da metodologia, algumas características dos modelos conceituais serão descritas e utilizadas para justificar a metodologia. As principais características da modelagem de sistemas apresentadas nesse trabalho são:

- **Modularização**

Os sistemas são decompostos em várias partes denominadas módulos ou blocos, cada um com características bem definidas. Os tipos de blocos empregados na metodologia, bem como suas características serão descritos mais adiante. Estruturalmente, estes blocos podem estar conectados serialmente, paralelamente ou em configuração que não esteja exatamente em conformidade com uma estrutura serial/paralela [140]. Um dos tipos de configuração paralela, denominada conexão m/n , representa um tipo especial de redundância paralela em que o bloco, ou o sistema, torna-se inoperante caso um mínimo de m componentes, de um total de n , estejam em falha. Ou seja, o bloco, ou o sistema, funcionará corretamente caso no mínimo $n-m+1$ componentes estejam funcionando convenientemente. Uma das formas de se representar o sistema por meio de um conjunto de blocos interconectados consiste na adoção de uma das técnicas combinatoriais, em especial na técnica de diagrama de blocos, bastante difundida nas diversas áreas da engenharia. Na metodologia desenvolvida, as conexões entre blocos são controladas por meio de parâmetros de configuração e parâmetros estruturais definidos nas redes de Petri que os representam. Os parâmetros de configuração, representados por lugares isolados denominados *flags*, definem os diferentes tipos de configurações dos blocos, enquanto os parâmetros estruturais definem a quantidade de blocos, as taxas de falha e de reparo, e os fatores de cobertura envolvidos.

- **Hierarquia**

A complexidade dos grandes sistemas pode ser de tal ordem que não permita uma análise desses sistemas como um todo. Desse modo, procura-se dividir e distribuir a complexidade do sistema em diversos níveis, cada qual com seu modelo associado, onde os resultados obtidos em um determinado nível são utilizados como entradas no nível seguinte. A medida que se passa de um nível de abstração para outro, detalhes são incorporados, através das metodologias de projeto dos tipos *bottom-up* e *top-down*, até a obtenção da especificação do sistema em um nível que permita estimativas de confiabilidade, disponibilidade e segurança do sistema como um todo. A idéia básica é o refinamento passo-a-passo de modelos de cenários complexos.

Inicialmente, cada bloco do diagrama do sistema é analisado e, se necessário, transformado em um modelo tolerante a falhas, o qual por sua vez é transformado em um diagrama de blocos de dependabilidade estendida (EDBD), pela incorporação de blocos

de decisão adicionais. Cada bloco do diagrama EDBD é transformado em dois modelos distintos: modelo de bloco intermediário e modelo EDSPN. Os modelos de blocos intermediários, representados por redes de Petri de alto nível, bem formadas [142], definem o diagrama de configuração do sistema por meio de regras de composição. Os modelos EDSPN, que representam o comportamento de cada bloco, são gerados e analisados individualmente, e os valores ou as expressões analíticas de dependabilidade obtidas são utilizadas como entrada do nível hierárquico superior formado por modelos dependáveis e parametrizados (MDP). Os modelos MDP, uma combinação dos modelos de blocos intermediários, que definem a configuração do sistema, com os valores ou expressões analíticas de dependabilidade de cada bloco, permitem a obtenção das estimativas de dependabilidade final do diagrama de sistema como um todo, de um modo rápido e sem que haja explosão de estados, conforme representação resumida mostrada na Figura 4.1.

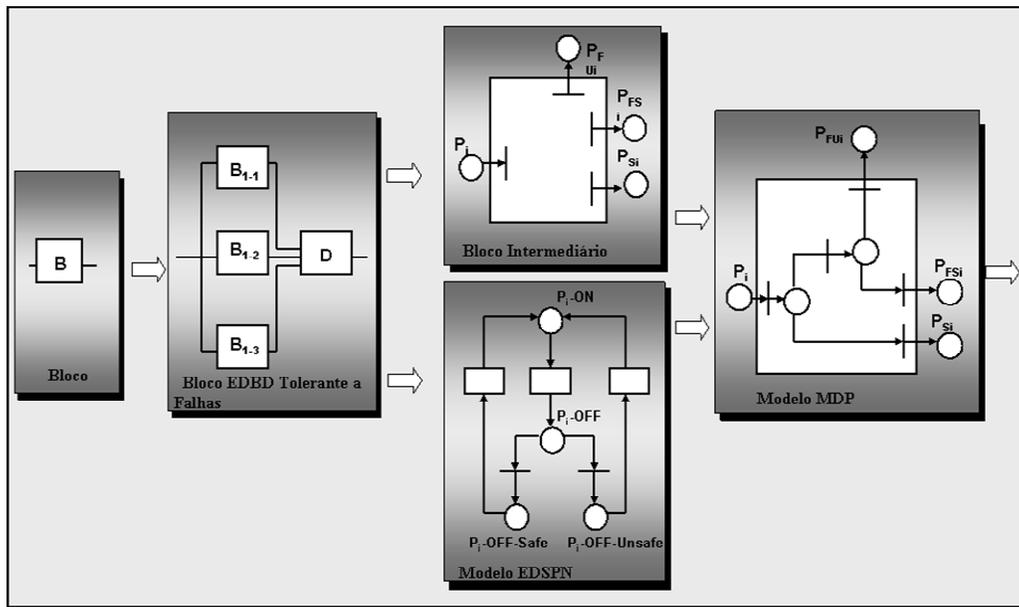


Figura 4.1 Evolução da metodologia: dos modelos de blocos aos modelos MDP.

4.1 Metodologia de Modelagem, Refinamento e Análise de Sistemas Dependáveis

A metodologia pode ser aplicada em duas situações distintas: a) aos sistemas já em funcionamento, como forma de eliminação de possíveis gargalos ou mesmo de aperfeiçoamento dos sistemas com relação aos aspectos de dependabilidade, através da análise e do refinamento dos modelos de alto nível; b) aos sistemas ainda na fase de projeto, através da comparação e análise das arquiteturas de projetos candidatos à implementação do sistema, e posterior refinamento da arquitetura selecionada.

Técnicas de tolerância a falhas utilizadas na modelagem de sistemas dependáveis objetivam garantir que os requisitos não-funcionais atribuídos aos sistemas, tais como confiabilidade, disponibilidade e segurança, sejam satisfeitos.

Diversos são os atores (profissionais) envolvidos num processo de avaliação dos sistemas dependáveis. Os insumos de entrada, os resultados gerados e as atividades desenvolvidas são definidos em relação às atribuições de cada ator no processo. Os três principais atores do processo são [54]:

- **Especialista:** é o ator que detém o conhecimento e as informações relevantes ao problema. A descrição do problema, seja em forma textual ou por meio de diagramas, as características técnicas dos dispositivos envolvidos no problema, a estrutura de interconexão desses dispositivos, os critérios de avaliação a serem utilizados ou mesmo as possíveis arquiteturas do sistema associadas ao problema, quando ainda em fase de projeto, são algumas de suas atribuições. A esse ator cabe ainda a função de analisar as implicações dos resultados obtidos dentro do escopo do problema, bem como os refinamentos e recomendações propostas para que possam ser atingidos os requisitos de dependabilidade do sistema;
- **Modelador:** é o ator que domina técnicas de representação dos sistemas por meio de modelos. Diagramas de bloco de confiabilidade, árvores de falha e especialmente redes de Petri estocástica são importantes métodos na representação dos sistemas tolerantes a falhas e no seu refinamento. Os modelos desenvolvidos por esse ator devem ser capazes de permitir a análise dos critérios definidos;
- **Avaliador:** é o ator que domina os métodos de avaliação utilizados. Dois tipos de soluções são de interesse quando se avalia as cadeias de Markov de tempo contínuo: transitória e de estado permanente. De modo a facilitar a obtenção de soluções transitórias por meio da resolução das equações diferenciais de Kolmogorov [140]], ou ainda soluções de estado permanente pela resolução de sistemas de equações lineares, ferramentas de modelagem de alto nível estão à disposição dos avaliadores. Dentre estas, utilizam-se os modelos de redes de Petri para avaliação de métricas, as quais subsidiarão o especialista com informações para tomadas de decisão.

4.2 Etapas da Metodologia

Para um melhor entendimento da metodologia de modelagem, refinamento e análise, uma divisão por fases é apresentada, onde cada fase constitui um conjunto de atividades, conforme pode ser visto na Figura 4.2.

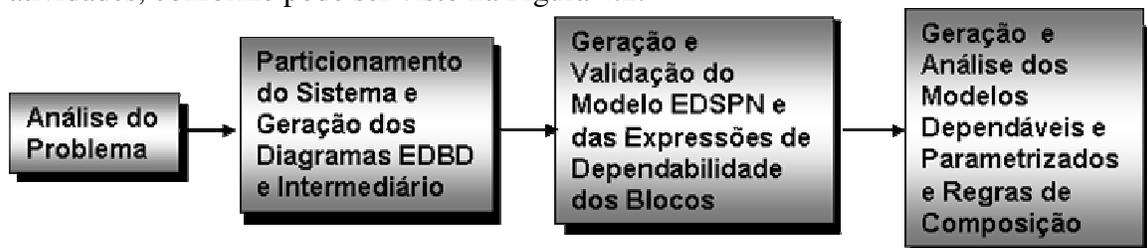


Figura 4.2 Fases de execução da metodologia de modelagem, avaliação e refinamento.

A metodologia de modelagem, cujo detalhamento será melhor explicitado quando de sua descrição geral, é apresentada de um modo mais abrangente na Figura 4.3. Antes, porém, de se especificar cada uma das fases de um modo mais detalhado, faz-se necessária uma breve explicação a respeito de cada uma das quatro fases mostradas na Figura 4.2:

- I. análise do problema:** esta fase objetiva uma definição clara e concisa do problema. Informações preliminares tais como, o tipo de tecnologia a ser utilizada, a apresentação de diagramas ou textos descrevendo as partes do sistema a serem projetadas e as interligações entre estas partes, podem ser utilizadas de modo a delimitar o escopo do problema. Nesta fase também são definidos os requisitos de dependabilidade a serem alcançados e o intervalo de tempo em que estes requisitos devem ser válidos.
- II. particionamento do sistema e geração dos diagramas EDBD e intermediário:** esta etapa compreende não apenas a segmentação da especificação do sistema em partes menores visando alcançar os requisitos de dependabilidade, baseados em critérios estruturais, funcionais ou de processamento, como também a transformação das partes críticas do diagrama por meio de técnicas de redundância e estratégias de tolerância a falhas, nos denominados diagramas de blocos de dependabilidade estendida (EDBD). Nesta fase, também são gerados os diagramas de blocos intermediários, os quais são redes de Petri cujos lugares e transições definem a interface de cada bloco com respeito às condições de disponibilidade, confiabilidade, falha segura e falha insegura.
- III. geração, análise, refinamento e validação dos modelos EDSPN:** esta fase consiste na geração e na análise qualitativa e quantitativa dos modelos EDSPN, com ou sem refinamentos, correspondentes a cada bloco do diagrama EDBD. Dados estatísticos ou do fabricante obtidos nesta fase relativos a partes do sistema, podem ser empregados na definição de médias e variâncias (desvio padrão), a serem utilizadas na validação dos modelos EDSPN e na determinação de valores numéricos ou expressões analíticas de dependabilidade, necessárias à fase seguinte.
- IV. geração e análise dos modelos dependáveis e parametrizados e regras de composição:** nesta fase são gerados os modelos MDP, os quais são representados por redes de Petri, conforme definido pelo diagrama de blocos intermediários e cujas transições são do tipo imediatas. Nos modelos MDP, os pesos das transições imediatas em conflito representam valores de probabilidade ou expressões analíticas de confiabilidade e disponibilidade, além de fatores de cobertura obtidos dos modelos EDSPN. O modelo MDP é dito parametrizado uma vez que por meio de regras estruturais definidas por expressões de multiplicidade de arcos dependentes das marcações e dos parâmetros de configuração, a serem definidos nos próximos capítulos, são estabelecidas formas de composição dos blocos intermediários na formação do diagrama de sistema dependável. Esta fase permite

a definição de métricas e métodos de avaliação do diagrama dependável de sistema.

Cada fase é composta por várias atividades, contidas nas elipses, as quais podem ser realizadas de um modo seqüencial ou paralelo. O processo mostrado na Figura 4.3, apesar de ser executado em seqüência, é também de natureza iterativa, à medida que alguns passos precisam ser revisitados em caso de inadequação dos resultados durante a análise.

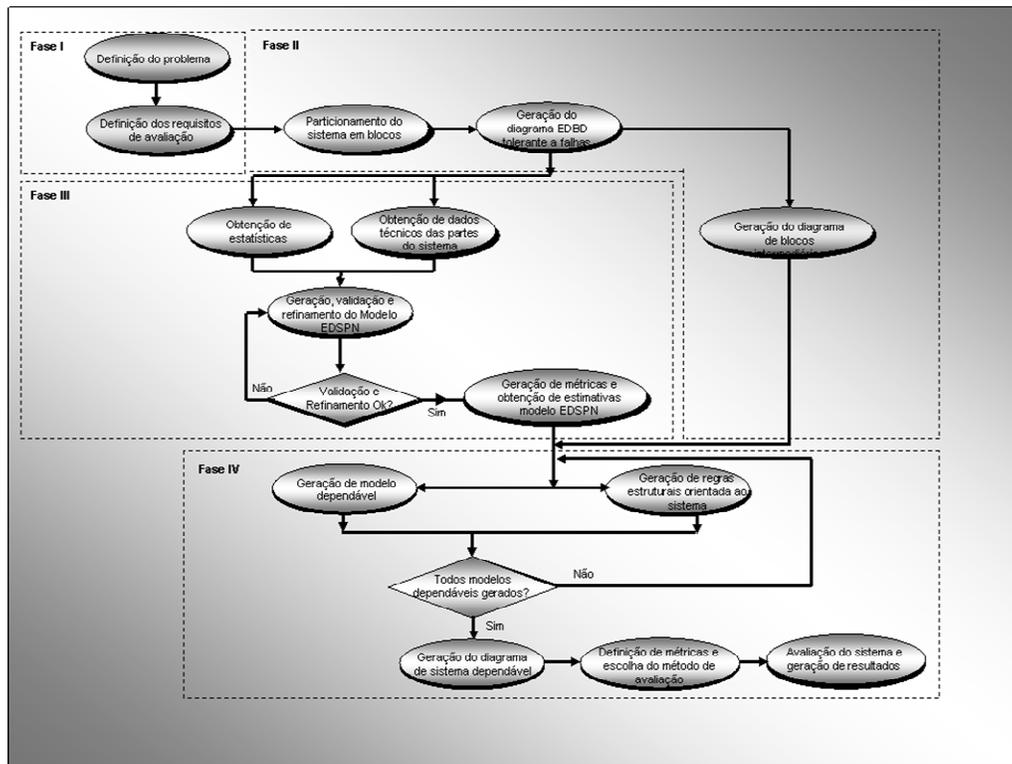


Figura 4. 3 Metodologia de modelagem, avaliação e refinamento de sistemas dependáveis

4.3 Metodologia – Descrição Geral

A seguir serão descritas cada uma das quatro fases que compõem o processo de modelagem, refinamento e análise de dependabilidade, conjuntamente com suas atividades, conforme mostrado na Figura 4.3.

- **Fase I: Análise do Problema**
- **Atividade: Definição do Problema**

Descrição: Esta atividade, desenvolvida pelo especialista, visa definir o escopo do problema de uma forma textual ou diagramática. Informações preliminares associadas ao problema, às necessidades a serem solucionadas e os objetivos a serem atingidos, também deverão ser objeto de descrição. Uma boa definição do problema evita interpretações ambíguas e descrições incompletas.

Entradas: Diagramas ou descrições textuais associados ao problema. Os diagramas podem representar um sistema de telecomunicações [54], um sistema eletrônico de controle de vôo de uma aeronave [70] ou um sistema de controle de disparo do motor de um foguete [141]. Características técnicas das partes do sistema que compõem o escopo do problema, de acordo com as necessidades e os objetivos definidos, também podem ser de importância nessa fase.

Saídas: Representação do problema em forma de um documento textual ou de diagrama composto pelas partes do sistema pertinentes ao escopo do problema, denominado diagrama do sistema. Uma listagem das partes do sistema que compõem o escopo do problema, suas funcionalidades e suas características com relação aos parâmetros MTTF, MTTR e MTBF, se houver, é produzida em documento, como resultado final dessa fase. As partes do sistema podem ser equipamentos, dispositivos, componentes ou um conjunto desses.

- **Atividade:** *Definição dos Requisitos de Avaliação*

Descrição: Esta atividade, desempenhada pelo especialista, objetiva a identificação dos requisitos ou critérios de avaliação pertinentes ao problema. No processo de modelagem e análise de dependabilidade, requisitos de confiabilidade, disponibilidade e segurança, gerados a partir das necessidades e objetivos inerentes à descrição do problema, definem a modelagem do sistema a ser adotada.

Entradas: Documento textual e diagramas contendo descrição detalhada do problema a ser solucionado, bem como informações preliminares dos parâmetros MTTF, MTTR e MTBF, se houver, além das características das partes do sistema contidas na listagem gerada na atividade anterior. Um outro item importante que deve ser estabelecido é o intervalo de tempo sobre o qual as operações do sistema devem ser garantidas. No caso dos sistemas de controle de vôo, o intervalo de tempo para validação dos requisitos não deve ser superior a quantidade de horas do vôo mais longo. No caso de um sistema de telecomunicações o intervalo de tempo pode ser definido como sendo de 1 ano, ou 8760 horas, como adotado em [55]. Um outro requisito válido é a quantidade de falhas suportadas pelo sistema, sem comprometimento de sua operação correta, ou ainda operações de falhas seguras após ocorrência de determinadas falhas. O intervalo de tempo aceitável entre a falha e a recuperação do sistema, de modo a mantê-lo com uma disponibilidade aceitável, é também de importância nessa etapa.

Saídas: Requisitos de confiabilidade, disponibilidade e segurança avaliados.

- **Fase II:** *Particionamento do Sistema e Geração do diagrama EDBD*
- **Atividade:** *Particionamento do Sistema em blocos*

Descrição: Esta atividade, realizada pelo modelador, consiste na transformação da especificação ou do diagrama geral do sistema gerado na fase anterior, por um diagrama de blocos, onde cada bloco corresponde a uma dada parte, ou várias partes do sistema, orientado aos requisitos de avaliação anteriormente definidos. O diagrama do sistema pode ser segmentado ainda de acordo com a sua configuração funcional, de acordo com a sua estrutura básica, ou conforme o modo de processamento, centralizado ou distribuído, por exemplo.

Entradas: Especificação ou diagrama geral do sistema e requisitos de avaliação.
Saída: Geração do diagrama de blocos do sistema.

- **Atividade:** *Geração do Diagrama de Blocos de Dependabilidade Estendida (EDBD)*

Descrição: Esta atividade, desenvolvida pelo modelador, transforma o diagrama de blocos do sistema em um diagrama EDBD. Os blocos desse novo diagrama podem ser os mesmos do diagrama anterior à transformação, i.e., blocos originais, ou podem ser novos blocos formados pela composição dos blocos originais com blocos redundantes adicionais, por meio de técnicas de redundância estáticas ou dinâmicas, de acordo com as estratégias de tolerância a falhas definidas, e blocos de decisão, se necessários, de modo a satisfazer os requisitos de dependabilidade do sistema no intervalo de tempo acordado.

Entradas: Diagrama de blocos do sistema e técnicas de tolerância a falhas.

Saída: Geração do diagrama EDBD.

- **Atividade:** *Geração do Diagrama de Blocos Intermediários*

Descrição: Esta atividade, realizada pelo modelador, transforma o diagrama EDBD de um sistema em um diagrama de blocos intermediário, formado por redes de Petri de alto nível de abstração e regras de composição estrutural dos tipos serial e paralela. A representação dos blocos intermediários, por meio das redes de Petri, permite a composição de um bloco com os demais blocos do diagrama, por meio de lugares e transições de entrada e saída. As regras de composição estrutural são regras formais que levam em conta os conceitos de fusão, junção e adição na descrição de sua composição [45]. Os blocos intermediários estendem os diagramas de blocos com respeito às análises de disponibilidade e segurança, além daquela de confiabilidade. Estes diagramas serão úteis na geração dos diagramas de sistemas dependáveis na fase final da metodologia

Entradas: Diagrama EDBD do sistema e regras de composição formal entre blocos representados por redes de Petri de alto nível bem formadas [142].

Saída: Diagrama de blocos intermediários.

- **Fase III:** *Geração e Validação do Modelo EDSPN e das Expressões de Dependabilidade dos Blocos.*
- **Atividade:** *Obtenção de Dados Estatísticos*

Descrição: Esta atividade, realizada pelo especialista e executada em sistemas já em funcionamento, objetiva a obtenção de parâmetros relacionados à ocorrência de falhas e reparos das partes que compõem o sistema, através de dados históricos coletados durante a sua operação real sob investigação, ou ainda a aquisição de dados de campo disponíveis de componentes de prateleira, comerciais ou não, para as arquiteturas candidatas de sistemas ainda na fase de projeto, ou ainda através de processos de amostragem. Os dados estatísticos considerados serão aqueles baseados no primeiro e segundo momentos, isto é, nas médias e variâncias (desvio padrão), respectivamente.

Entradas: Dados históricos de falhas e reparos das partes do sistema correspondentes aos blocos do diagrama de blocos do sistema

Saídas: Médias (MTTF, MTBF, MTTR), variâncias, desvios padrões e valores de cobertura de falhas das partes do sistema, representadas por blocos no diagrama de blocos do sistema.

- **Atividade:** *Obtenção de Dados Técnicos das Partes do Sistema*

Descrição: Esta atividade, executada pelo especialista, visa obter as características técnicas das partes de um sistema representadas por equipamentos, dispositivos e componentes. Um equipamento pode ser definido como um conjunto estruturado de dispositivos, enquanto que um dispositivo pode ser definido como um conjunto estruturado de componentes. Estes dados técnicos complementares são necessários para que se tenha um conhecimento ainda mais detalhado de cada parte do sistema, o qual é representado por um diagrama de blocos, caso os dados históricos pertinentes ao sistema não sejam suficientes ou mesmo não existam.

Entradas: Diagrama de blocos do sistema e especificações técnicas dos equipamentos, dispositivos e componentes do sistema fornecidos nos manuais dos fabricantes.

Saídas: Taxas de falha, tempos médios de falha e de reparo, MTTF e MTBF, respectivamente, dos equipamentos, dispositivos e componentes do sistema.

- **Atividade:** *Geração dos Modelos EDSPN para cada Bloco, ou Conjunto de Blocos, Isoladamente.*

Descrição: Esta atividade, desenvolvida pelo modelador, objetiva a transformação de cada bloco ou conjunto de blocos de um diagrama EDBD em modelos formais individuais do tipo EDSPN, gerados e analisados isoladamente e, em geral, independentemente, de modo a permitir a análise de dependabilidade das partes de um sistema, representada por esses blocos. Nesta atividade também são geradas as regras de composição dos blocos ou do conjunto de blocos. As regras no modelo EDSPN podem ser organizadas segundo: a) estruturas de configuração, as quais podem ser serial, paralela, m/n, NMR, entre outras; b) métodos de avaliação numéricos ou analíticos; c) avaliações transitórias ou de estado permanente; d) requisitos de dependabilidade, tais como confiabilidade, disponibilidade e segurança.

Em seguida é executada a análise qualitativa do modelo EDSPN e processada a sua validação. O processo de validação é realizado formalmente com o auxílio de ferramentas computacionais através de recursos de modelagem, dentre os quais o recurso conhecido como *token game*, de muita utilidade prática. Após o processo de geração e validação, pode-se refinar o modelo pela transformação de lugares, transições e arcos, pela introdução de funções lógicas condicionais dependentes da marcação e de parâmetros de configuração. O refinamento possibilita a representação de distribuições não-Markovianas, por meio de distribuições exponenciais em fases, a representação de diferentes disciplinas de escalonamento e políticas de reparo, ou ainda a representação de modelos em hardware e software.

Entrada: Conjunto de blocos pertencentes ao diagrama EDBD do sistema, configuração estrutural dos blocos, tipos e métodos de avaliação.

Saída: Modelo EDSPN correspondente a cada bloco, ou conjunto de blocos, do diagrama EDBD do sistema, cujas regras de composição estão associadas à estrutura, ao tipo de avaliação e ao método de avaliação de cada bloco.

- **Atividade:** *Geração de Métricas, Obtenção das Expressões dos Modelos EDSPN*

Descrição: Esta atividade, executada pelo avaliador, define funções analíticas ou valores numéricos para as estimativas de confiabilidade, disponibilidade e segurança, considerando-se os requisitos estabelecidos durante a Fase I, para cada bloco, ou conjunto de blocos, isoladamente. Esta atividade leva em conta ainda os refinamentos produzidos e as taxas ou tempos médios de falha e de reparo obtidos em atividades da Fase III e as regras de composição estrutural orientada a blocos. Esta atividade possibilita a definição de métricas de dependabilidade e a obtenção de expressões numéricas ou analíticas, as quais serão utilizadas pelos modelos MDP, em fase posterior.

Entradas: Modelo EDSPN, taxas ou tempos médios de falha e reparo e fatores de cobertura de cada bloco ou conjunto de blocos obtidos dos dados estatísticos e das características técnicas das partes do sistema, além das regras estruturais orientadas a blocos e aos requisitos definidos.

Saídas: Métricas representadas por expressões lógicas dependentes de marcação relativas aos atributos de confiabilidade, disponibilidade e segurança de cada bloco, e expressões numéricas ou analíticas para utilização nos modelos MDP.

- **Fase IV:** *Geração de Modelos Dependáveis e Parametrizados e Regras de Composição*
- **Atividade:** *Geração de Modelos Dependáveis e Parametrizados (Modelos MDP)*

Descrição: Esta atividade, desenvolvida pelo modelador, define a construção dos modelos MDP de alto nível para os blocos do diagrama EDBD, baseado nas expressões numéricas ou analíticas de dependabilidade, obtidas para cada um dos blocos, ou conjuntos de blocos, através dos modelos EDSPN, na fase anterior, e na configuração dos blocos que constituem o diagrama de blocos intermediário. Nesta fase também são definidos os parâmetros de configuração e estruturais para os blocos do diagrama intermediário, de acordo com o tipo de redundância associada aos blocos no diagrama EDBD. As expressões analíticas ou numéricas, obtidas dos modelos EDSPN, e definidas nos modelos MDP, estão relacionadas às funções de peso das transições imediatas conflitantes as quais são do tipo falha ou sucesso, e detecção ou não-deteção de falha.

Entradas: Diagrama EDBD do sistema, diagrama de blocos intermediário, expressões numéricas e/ou analíticas de confiabilidade, disponibilidade e segurança obtidas dos modelos EDSPN da fase anterior, e parâmetros de configuração e estruturais dos modelos MDP associados a cada um dos blocos do diagrama de blocos intermediário do sistema.

Saídas: Geração dos modelos MDP.

- **Atividade:** *Definição de Regras Estruturais Orientadas ao Sistema*

Descrição: Esta atividade, desenvolvida pelo modelador, estabelece as expressões lógicas condicionais necessárias à composição dos modelos MDP, correspondentes aos blocos do diagrama de blocos intermediário do sistema definido na Fase II. Algumas regras de composição entre modelos MDP são semelhantes aquelas dos modelos EDSPN orientada a blocos e definidas na Fase III. As regras de composição estrutural dependem da estrutura dos blocos no diagrama de blocos intermediário, conforme definida no diagrama EDBD na Fase II, as quais poderão ser serial, paralela, *m/n*, TMR, NMR e *flux-summing*.

Entradas: Diagrama EDBD, diagrama de blocos intermediário, parâmetros de configuração e parâmetros estruturais.

Saídas: Geração dos modelos MDP, com as respectivas regras de decisão associadas à estrutura dos blocos no diagrama de blocos intermediário.

- *Atividade: Geração do Diagrama de Sistemas Dependável*

Descrição: Esta atividade, realizada pelo modelador, define a construção de um diagrama de sistema dependável, de alto nível, baseado nos modelos MDP dos blocos que o compõem, e nas regras de composição orientadas ao sistema definidas anteriormente.

Entradas: Modelos MDP e regras de composição orientada ao sistema.

Saídas: Diagrama de Sistema Dependável.

- *Atividade: Definição de Métricas e Escolha do Método de Avaliação*

Descrição: Esta atividade, desenvolvida pelo avaliador, define as métricas a serem geradas para obtenção das estimativas de dependabilidade do sistema como um todo, além de definir a técnica de avaliação estocástica a ser utilizada, i.e., análise ou simulação estocástica, além dos parâmetros e opções associadas como, por exemplo, grau de precisão, tempo de análise, método de solução, intervalo de confiança desejado, entre outros.

Entradas: Diagrama de sistema dependável.

Saída: Documento de Avaliação contendo as métricas e os parâmetros e opções da técnica de avaliação estocástica a ser utilizada.

- *Atividade: Avaliação do Sistema e Geração de Resultados*

Descrição: Esta atividade, desempenhada pelo avaliador, executa os passos definidos no documento de avaliação através da ferramenta de modelagem em uso.

Entradas: Documento de avaliação e diagrama de modelos dependáveis.

Saída: Valores de dependabilidade do sistema.

4.4 Níveis Hierárquicos

Os grandes sistemas podem apresentar uma complexidade de tal ordem que se torna difícil ou mesmo impossível a sua análise de uma forma monolítica. Uma das soluções para se lidar com tal complexidade consiste na adoção de metodologias modulares e

hierárquicas. Uma vez que o sistema pode ser contextualizado como um conjunto de blocos ou subsistemas menores, o modelador poderá focar em cada um dos blocos por vez, simplificando dessa forma não apenas a compreensão da funcionalidade do sistema como um todo, como também o desenvolvimento do processo de modelagem. Considera-se, na metodologia desenvolvida para o processo de modelagem, refinamento e análise de dependabilidade, que cada bloco representa uma determinada parte do sistema e cada um desses blocos pode ser composto por vários outros blocos representando sub-partes, e assim por diante, pode-se decompor a metodologia, de um modo hierárquico, em cinco grandes níveis:

- **nível hierárquico 1:** diagrama de blocos do sistema;
- **nível hierárquico 2:** diagramas de blocos de dependabilidade estendida (EDBD);
- **nível hierárquico 3:** diagrama de blocos intermediários;
- **nível hierárquico 4:** modelos EDSPN;
- **nível hierárquico 5:** diagrama de sistema dependável e parametrizado;

4.4.1 Nível Hierárquico 1: Diagrama de blocos do sistema

Este nível hierárquico envolve a etapa inicial da metodologia, a qual de acordo com a situação a que se destina, poderá ter duas abordagens distintas:

- **Sistema em funcionamento:**

Caso o sistema já esteja implantado, esta fase da metodologia propõe um completo entendimento do sistema e a obtenção dos valores de dependabilidade para avaliação de possíveis gargalos. Esse entendimento do sistema é feito por intermédio de um levantamento de todo o material textual ou diagramático existente sobre o sistema a ser avaliado. Neste nível, são definidos os critérios de dependabilidade a serem avaliados, os valores a serem obtidos e os elementos que compõem o escopo do sistema. A infraestrutura de comunicação utilizada como suporte de automação a um sistema de transmissão de energia elétrica, por exemplo, pode ser composta pelos diagramas dos sistemas de telefonia e das redes WAN, e pelos diagramas dos sistemas de teleproteção e de transmissão por eles suportados, o que torna o sistema bastante complexo e de difícil modelagem. É necessário, portanto, a definição do problema com base nos parâmetros previamente estabelecidos dentro de um escopo bem delimitado que torne possível a construção do modelo e sua análise.

- **Sistema em fase inicial de projeto:**

Caso o sistema ainda esteja na fase de projeto, esta fase consiste na definição do problema, de uma forma clara e concisa, e na definição de um conjunto de requisitos não-funcionais que se deseja atender, os quais, na metodologia, estão relacionados à

confiabilidade, disponibilidade e segurança. Neste nível, deseja-se observar, dentre os vários modelos das arquiteturas candidatas àquela que mais se aproxima dos requisitos de dependabilidade propostos durante a especificação. Como o sistema ainda não está implementado, não é possível a obtenção de estimativas de tempos médios e taxas, como no caso anterior, com o sistema já em funcionamento. Portanto, as arquiteturas propostas devem conter componentes que estejam sendo utilizados em outros sistemas já em funcionamento, de modo que se possa obter estimativas de dependabilidade, ou ainda a utilização de dispositivos COTS (*Commercial Off The Shelf*), para os quais dados de campo referentes aos atributos de dependabilidade podem estar disponíveis. Estando estas arquiteturas candidatas na fase inicial de projeto, pode-se ainda assumir valores baseados na experiência adquirida em experimentos anteriores, caso não haja disponibilidade de valores reais dos parâmetros dos seus componentes. Nesse caso, a medida que novos conhecimentos do sistema são adquiridos, pode-se refinar o modelo, de modo a melhorar a qualidade dos parâmetros arbitrados.

Os requisitos de dependabilidade extraídos da descrição do problema devem ser passíveis de obtenção analítica ou experimental, isto é, devem estar dentro de limites coerentes ou realistas. Caso contrário, os requisitos propostos poderão não ser passíveis de obtenção ou, ainda que o sejam, poderão ter um custo de obtenção extremamente elevado [70]. Uma vez que os requisitos estejam bem compreendidos e definidos, deve-se especificar o sistema por partes, de modo a reduzir a complexidade da sua avaliação. A especificação do sistema é representada por meio de um diagrama de blocos, conforme mostrado na Figura 4.4, onde cada bloco representa uma parte do sistema. As partes de um sistema podem ser, por exemplo, um equipamento, uma funcionalidade ou unidades de hardware e software. O processo de particionamento poderá ainda ser baseado nos requisitos de dependabilidade, considerando-se, por exemplo, diferentes níveis de criticidade dos blocos. A indústria aeronáutica, que utiliza este critério de particionamento, classifica as funções do sistema como sendo funções críticas de voo, funções críticas de missão e funções convenientes, as quais têm diferentes requisitos de confiabilidade [70].

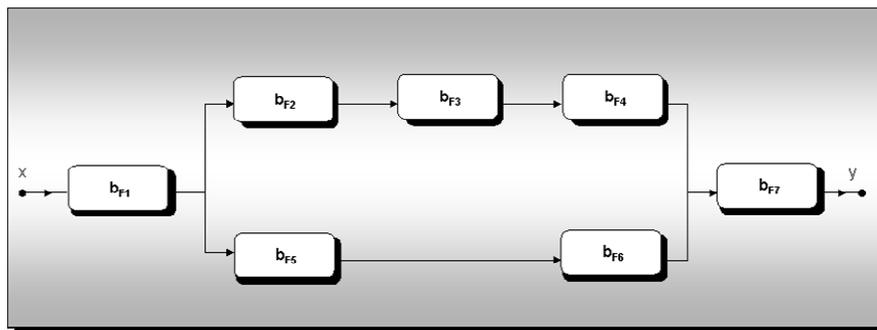


Figura 4.4 Especificação do Sistema em Diagrama de Blocos

4.4.2 Nível Hierárquico 2: Diagramas de blocos de dependabilidade estendida (EDBD)

Para que se possa reduzir a complexidade da análise dos diagramas de blocos tolerantes à falhas de um sistema definidos na Fase II da metodologia, algumas premissas devem ser consideradas, conforme definições a seguir:

- assim como nos modelos de diagrama de blocos confiáveis (RBD), o sistema representado pelo diagrama de blocos tolerantes a falhas funcionará coerentemente, ou seja, será confiável, se todos os blocos que compõem pelo menos um dos caminhos entre os pontos terminais x e y não apresentarem falha [144];
- só será considerada a ocorrência de uma falha por vez, isto é, mais de uma falha ocorrendo ao mesmo tempo no mesmo bloco ou em blocos distintos não serão consideradas;
- o bloco ao ser recuperado será considerado como se fosse novo, de novo;
- as técnicas de tolerância a falhas atribuídas aos blocos críticos devem ser classificadas de acordo com a condição dos blocos redundantes, definidos como réplicas, descritos mais adiante;
- o bloco votador, responsável pelo mecanismo de votação das técnicas NMR é considerado perfeito, caso contrário a sua dependabilidade deve ser levada em conta;
- os fatores de cobertura são considerados conhecidos para cada modo de falha.

Antes da definição dos blocos que compõem o modelo EDBD, se faz necessário a definição das formas de replicação utilizadas. Define-se réplica como sendo o conjunto de dois ou mais blocos com a mesma funcionalidade. As réplicas podem representar dispositivos extras de hardware ou variantes de software baseadas numa mesma especificação.

Várias são as técnicas de replicação utilizadas nas diversas arquiteturas, conforme descrito em [109][1]. A estas técnicas será acrescentada a replicação passiva com réplica inativa sujeita à falha. As várias formas de replicação são descritas como:

- **replicação passiva – réplica inativa livre de falha:** neste tipo de replicação apenas uma das n réplicas do modelo redundante irá processar os dados ou mensagens de entrada fornecendo um resultado de saída. As demais réplicas não processarão os dados de entrada e nem estarão sujeitas a falhas nesse período, permanecendo desenergizadas. Caso a réplica ativa torne-se inabilitada, esta será substituída por uma das réplicas em espera que se encontra inativa. Este modo de replicação será utilizado na técnica de tolerância a falhas *coldstandby sparing*, a ser definido mais adiante. Um dos inconvenientes desse tipo de replicação ocorre quando da troca do bloco ativo por um dos blocos passivos em espera. O conjunto de réplicas fica desabilitado até que o bloco em espera seja inserido, energizado e inicializado, após restauração do contexto onde ocorreu a falha do bloco ativo. Essa troca pode ser otimizada pelo uso da técnica de *hot swap* (troca a quente), onde o bloco faltoso é substituído sem que haja necessidade de desenergização.

- **replicação passiva - réplica inativa sujeita a falha:** neste caso, assim como no caso anterior, apenas uma das n réplicas irá processar os dados ou mensagens de entrada e fornecer resultados para a saída. As demais réplicas não processarão os dados de entrada, porém estarão sujeitas a falhas nesse período de inatividade, com taxas de falha em geral menor do que a taxa de falha quando ativa. Caso a réplica ativa torne-se inabilitada será substituída por uma das réplicas inativas. A réplica em espera ao assumir a condição ativa, passará a ter a taxa de falha da condição ativa. Esse modo de replicação será utilizado na técnica de tolerância a falhas *warmstandby sparing*;
- **replicação semi-ativa:** neste caso, todas as réplicas do sistema estarão ativas, processando simultaneamente as mesmas entradas, com o resultado de saída, contudo, sendo fornecido por apenas uma das réplicas, a réplica primária. Neste tipo de replicação, a réplica primária e as réplicas secundárias se encontram permanentemente energizadas e prontas para serem comutadas em caso de necessidade. Caso a réplica primária ativa venha a falhar, esta será comutada por uma das réplicas secundárias, que passará a ser primária. Este modo de replicação será utilizado na técnica de tolerância a falhas *hotstandby sparing*.
- **replicação ativa NMR:** neste modo de replicação, todas as réplicas estão ativas, energizadas e sujeitas a falhas, processando concorrentemente as entradas e liberando de uma forma síncrona os resultados para um mecanismo de decisão, em geral um votador. Neste tipo de replicação, o número de réplicas n é um número inteiro e ímpar, e um total de até $(n-1)/2$ réplicas em falha pode ser suportado por meio de mascaramento. Um resultado válido será liberado quando pelo menos $(n+1)/2$ réplicas liberarem resultados válidos. Este modo de replicação ativa será utilizado na técnica de tolerância a falhas NMR (*N- Modular Redundancy*).
- **replicação ativa generalizada:** a replicação ativa tem na configuração m/n o seu caso mais geral. Nesta configuração, caso o sistema possua um mínimo de m das n réplicas numa condição de falha, o sistema ficará inativo. Logo, para que o sistema funcione convenientemente ele deverá ter no mínimo $n-m+1$ componentes funcionando corretamente. Assim, como na replicação ativa NMR, todas as réplicas estão ativas e energizadas, concorrentemente processando as entradas e liberando os resultados de forma síncrona a um mecanismo de decisão.

De modo a facilitar o processo de modelagem de alto nível e de permitir a análise dos requisitos do sistema, além de uma maior compactação, nas dimensões do diagrama de blocos, tolerante a falhas, os seguintes blocos são definidos:

- **bloco básico funcional** é aquele que não apresenta redundância e cuja ocorrência de falhas obedece a uma distribuição exponencial (Markoviana);
- **bloco ativo** é o que utiliza técnicas de redundância estática para incrementar o seu nível de dependabilidade. As técnicas de redundância estática consideradas são do tipo: a) TMR (*Triple Modular Redundancy*), caso particular da técnica NMR (*N- Modular Redundancy*) para um número de réplicas igual a 3; b) NMR, para um

número de réplicas qualquer, como por exemplo, TMR, 5MR, 7MR, 11MR, etc., *Flux-summing*, caso particular de TMR, usada basicamente em sistemas de aeronaves;

- **bloco *standby*** é o que utiliza técnicas de redundância dinâmica para incrementar o nível de dependabilidade. As técnicas de redundância dinâmica consideradas são dos tipos: a) *coldstandby* - uma das réplicas é ativa e sujeita a falhas, e as demais réplicas são passivas, não sujeitas a falhas; b) *warmstandby* - a réplica ativa e as réplicas passivas estão sujeitas a falhas com taxas de falhas diferentes entre a condição ativa e passiva; c) *hotstandby* – todas as réplicas estão ativas e sujeitas a falhas com a mesma taxa de falha, porém só a réplica primária fornece os resultados.
- **bloco subsistema** é aquele formado por parte da estrutura do diagrama de blocos de um sistema, isto é, composto por dois ou mais blocos interconectados e que pode ser reusado no mesmo diagrama de blocos do sistema ou em diagramas de blocos de outros sistemas. A distribuição de falhas poderá obedecer a uma distribuição qualquer. Em geral utiliza-se o bloco de subsistema quando se deseja representar uma estrutura de blocos com distribuição de probabilidade complexa por meio de avaliações numéricas.
- **bloco serial múltiplo** é o formado por dois ou mais blocos em série, que tem como finalidade tornar o diagrama mais compacto. Dois blocos estão em série quando a saída de um bloco é a única entrada do próximo;
- **bloco básico de decisão** é o bloco livre de falhas colocado externamente na saída de blocos paralelos de modo a liberar uma saída válida ou inválida, de acordo com regras lógicas de comutação associadas a um votador perfeito. Blocos estão em paralelo quando as entradas desses blocos provêm de blocos comuns, as operações são executadas concorrentemente, e as saídas desses blocos também se destinam a blocos comuns. Blocos paralelos, no contexto desta Tese, apenas admitem como restrição os resultados de saída serem liberados para blocos comuns. O bloco de decisão, quando definido dentro de um determinado bloco, configura o modo como os diversos blocos componentes desse bloco devem estar conectados de modo a liberar resultados válidos ou inválidos. As configurações intra-blocos ou inter-blocos dos tipos serial, paralelo, m/n, NMR e *flux-summing* são definidas por meio de expressões lógicas condicionais dependentes da marcação e dos parâmetros de configuração e estruturais. Definições formais de conexões de blocos paralelos e seriais podem ser obtidas em [7].

Os blocos básicos ou primitivos são definidos como os blocos que não podem ser decompostos em outros blocos. Os blocos funcionais e os blocos de decisão são exemplos de blocos básicos. Por outro lado, blocos compostos são definidos como os blocos constituídos por um conjunto de blocos básicos ou mesmo compostos. Os blocos compostos são representados pelos blocos ativos (ou múltiplos blocos em paralelo), pelos blocos *standby* (*cold*, *warm* ou *hot*), pelos blocos de subsistema e pelos blocos seriais

múltiplos. Os blocos compostos contêm, em sua estrutura, mais de um bloco, alguns dos quais, réplicas.

Considere-se, por exemplo, uma configuração arbitrária de sistema, onde quatro dos sete blocos em operação são considerados críticos, conforme pode ser observado nas Figuras 4.4 e 4.5. Os blocos b_{F3} , b_{F5} , b_{F6} e b_{F7} mostrados na Figura 4.4 são substituídos por blocos tolerantes a falhas na Figura 4.5 de modo a aumentar o nível de dependabilidade do sistema. O bloco crítico b_{F3} é substituído por um bloco tolerante a falhas que utiliza a técnica *coldstandby sparing* constituído por três blocos, uma réplica ativa e duas réplicas passivas; os blocos críticos b_{F5} e b_{F6} são substituídos por blocos tolerantes a falhas que utilizam a técnica *hotstandby sparing*, constituídos por três réplicas ativas e duas réplicas ativas, respectivamente; o bloco crítico b_{F7} é substituído pelo bloco tolerante a falhas TMR, composto por 3 blocos redundantes ativos, mais o bloco votador, responsável pelo processo de decisão.

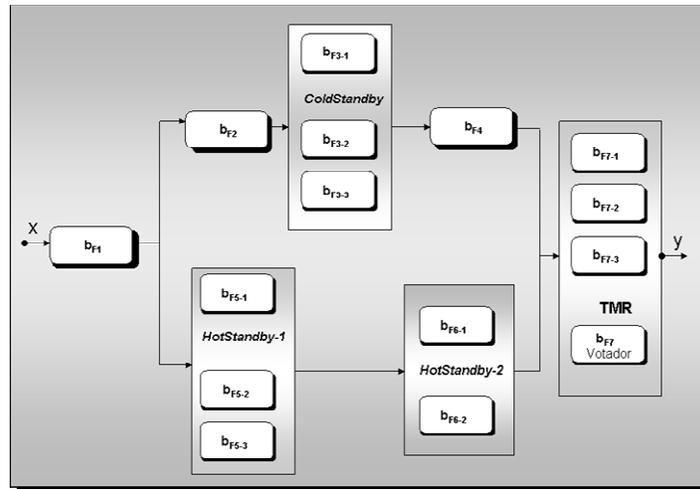


Figura 4.5 Diagrama de blocos tolerante a falhas

A especificação de um sistema, de acordo com [70], é um plano detalhado de projeto capaz de satisfazer determinados requisitos ou critérios. A introdução de técnicas de tolerância a falhas, nas metodologias de modelagem, torna-se necessária, caso se deseje atingir altos níveis de dependabilidade. As técnicas de tolerância a falhas *coldstandby*, *warmstandby* e *hotstandby* são técnicas de redundância estáticas, as quais conduzem o bloco a um processo de reconfiguração, quando é detectada falha na réplica ativa, e esta é comutada por uma das réplicas inativas em espera. Nas técnicas que utilizam replicação ativa, como TMR, por exemplo, não há necessidade de detecção de falha de qualquer das réplicas ativas, uma vez que os erros provenientes de falhas das réplicas são mascarados, ou seja, não são detectados.

Conforme pode ser observado na Figura 4.6, o bloco de decisão b_{D1} é introduzido para permitir o recebimento dos resultados dos blocos paralelos b_{F4} e *hotstandby-2*, e propiciar a geração de um resultado de saída para o bloco TMR seguinte. Caso se deseje compactar o diagrama de blocos tolerante a falhas, podem-se substituir os blocos b_{F2} , *coldstdby* e b_{F4} , por um bloco serial múltiplo. Caso se deseje compactar ainda mais o

diagrama e reusar a estrutura de blocos pertencentes aos caminhos paralelos na Figura 4.6, pode-se utilizar um bloco de subsistema. Os blocos pertencentes a Figura 4.6 são analisados e validados de um modo isolado, o que facilita a análise e minimiza a possibilidade de explosão de estados.

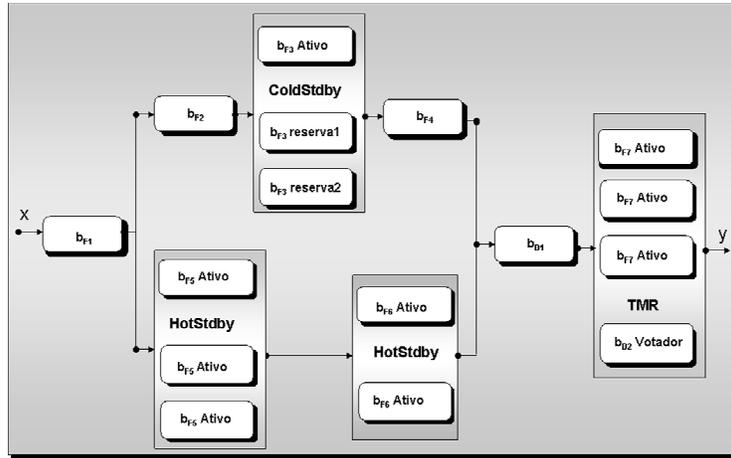


Figura 4.6 Diagrama de blocos tolerante a falhas com bloco de decisão

O diagrama tolerante a falhas da Figura 4.6, denominado diagrama de blocos de dependabilidade estendida EDBD, é composto por um grafo direcionado (B,A,x,y) , onde: $B=\{b_{k1}, b_{k2}, b_{k3}, \dots, b_{knk}\}$ é o conjunto de blocos que compõem o diagrama EDBD e $k \in \{f,a,d,r,m,s\}$ é o índice que define o tipo de bloco:

- f = bloco funcional;
- v = bloco ativo;
- d = bloco de decisão;
- r = bloco *standby*;
- m = bloco serial múltiplo;
- s = bloco de subsistema.

e $n_k \in \{n_f, n_v, n_d, n_r, n_m, n_s\}$ é o sub-índice que representa o número de blocos do tipo k no diagrama EDBD.

Um diagrama de blocos EDBD é constituído por um conjunto de blocos B , onde $B=B_f \cup B_v \cup B_d \cup B_r \cup B_m \cup B_s$ é a união dos diversos tipos de blocos no diagrama EDBD. $A=\{a_{(b_{ki}, b_{kj})}\}$ é um conjunto de arcos direcionados, sendo cada arco associado a um par ordenado (b_{ki}, b_{kj}) de blocos adjacentes, onde o bloco b_{ki} é o bloco fonte do arco, o bloco b_{kj} é o bloco destino e $\{(b_{ki}, b_{kj})\} \subseteq (B,B)$. Um arco nulo corresponde a não existência de arco direcionado entre um par de blocos.

Os elementos x e y , do diagrama EDBD, são nós *dummy* utilizados apenas para estabelecer um caminho de dependabilidade no sistema. Se, a qualquer instante de tempo, houver um caminho através do sistema do nó *dummy* x ao nó *dummy* y , então o sistema será considerado operacional; caso contrário, o sistema será considerado falho.

O conjunto $B_f = \{b_{f1}, b_{f2}, b_{f3}, \dots, b_{fn_f}\}$ representa os blocos funcionais básicos e $n_f \in \mathbb{N}$ representa a quantidade de tais blocos básicos no diagrama EDBD.

O conjunto $B_d = \{b_{d1}, b_{d2}, b_{d3}, \dots, b_{dn_d}\}$ representa os blocos de decisão básicos encontrados no diagrama EDBD, e $n_d \in \mathbb{N}$ representa a quantidade total desses blocos de decisão no diagrama EDBD. Os blocos de decisão recebem os resultados dos blocos de entrada a eles conectados e libera um resultado para os blocos de saída também a eles conectados por meio de regras de decisão.

O conjunto de blocos *standby*, $B_r = \{b_{r1}, b_{r2}, b_{r3}, \dots, b_{rn_r}\}$, representa todos os blocos tolerantes a falhas do tipo *coldstandby*, *warmstandby* ou *hotstandby* encontrados no diagrama EDBD, e $n_r \in \mathbb{N}$ representa a quantidade total desses blocos. Cada bloco *standby* é composto por duas ou mais réplicas, ativas e/ou passivas.

O conjunto de blocos ativos, $B_v = \{b_{v1}, b_{v2}, b_{v3}, \dots, b_{vn_v}\}$, representa os blocos que utilizam técnicas de tolerância a falhas do tipo *m/n* ou NMR, encontradas no diagrama EDBD, e $n_v \in \mathbb{N}$ representa a quantidade total desses blocos. Cada bloco NMR ou *m/n* é composto por um bloco votador e várias réplicas, sendo o número de réplicas um número natural e ímpar no caso NMR, e um número qualquer no caso *m/n*.

O conjunto $B_m = \{b_{m1}, b_{m2}, b_{m3}, \dots, b_{mn_m}\}$, representa todos os blocos múltiplos encontrados em um diagrama EDBD, cujo total é limitado por $n_m \in \mathbb{N}$. Individualmente, cada bloco múltiplo representa blocos funcionais em série, adjacentes ou não, com a mesma ou diferentes funcionalidades, cuja função é propiciar redução de espaço e de tempo em um diagrama EDBD.

O conjunto formado por $B_s = \{b_{s1}, b_{s2}, b_{s3}, \dots, b_{sns}\}$ representa todos os blocos de subsistemas contidos em um diagrama EDBD, limitado a n_s , onde $n_s \in \mathbb{N}$. Cada bloco de subsistema contém uma parte do sistema, isto é, contém blocos e os respectivos arcos adjacentes que os interligam. A representação de parte de um diagrama EDBD por um bloco único de subsistema, além de propiciar uma maior flexibilidade, simplifica a representação do sistema em termos de blocos e permite a reutilização desse bloco tantas vezes quantas forem necessárias no diagrama EDBD do sistema.

A descrição por meio de diagrama de blocos facilita o entendimento do sistema uma vez que divide a complexidade de sua análise através dos vários blocos que o compõe. A análise do sistema por meio de um diagrama EDBD, por outro lado, apesar de oferecer a possibilidade de incorporação de técnicas de tolerância a falhas aos componentes críticos de um sistema, não detalha a conexão entre esses blocos, nem como as condições lógicas de dependabilidade, representadas pelas condições sucesso e falha, ativo e inativo, operacional e não-operacional, podem ser definidas entre eles. A incorporação dessas condições ao diagrama de blocos produz um novo diagrama, denominado diagrama de blocos intermediários, com um nível maior de detalhes.

4.4.3 Nível Hierárquico 3: Diagrama de blocos intermediários

O diagrama de blocos intermediário permite o refinamento do diagrama EDBD por meio de uma rede de Petri de alto nível de abstração caracterizada por lugares e transições que compõem a interface do bloco. O diagrama de blocos intermediários é constituído por dois tipos de blocos:

a) **blocos intermediários:** são blocos que representam os blocos funcionais básicos ou blocos compostos, e que são representados por redes de Petri compostas por um lugar e uma transição de entrada, e de três lugares de saída, com suas respectivas transições, representando as condições lógicas de saída do bloco, conforme a Figura 4.7;

b) **blocos de decisão intermediários:** são blocos básicos de decisão, representados por redes de Petri compostas por três lugares de entrada com suas respectivas transições, representando as condições lógicas dos blocos de entrada, e por três lugares de saída com suas respectivas transições, representando a combinação dos resultados lógicos obtidos dos blocos de entrada de acordo com regras de decisão, conforme a Figura 4.8.

Nas Figuras 4.7 e 4.8, o lugar de saída M_{Si} representa a condição de sucesso na execução do bloco; o lugar M_{FUi} representa a condição de falha insegura, isto é, falhas não passíveis de detecção; o lugar M_{FSi} representa a condição de falha segura, isto é, falhas capazes de serem detectadas. Na Figura 4.8, as marcações dos lugares P_{FUi} , P_{FSi} e P_{Si} , definidas por n_1 , n_2 e n_3 representam a quantidade de blocos intermediários de entrada nas condições de falha insegura, falha segura e sucesso, respectivamente, e os lugares de saída M_{FUi} , M_{FSi} e M_{Si} representam os resultados exclusivos da decisão de falha insegura, falha segura ou sucesso, pela aplicação de regras de composição aos blocos de entrada. Como a representação do sistema, por meio de um diagrama de blocos intermediários, segue o conceito de modularização, não apenas a especificação de cada bloco intermediário deve ser definida, como também a sistematização de procedimentos de composição desses blocos. A composição dos blocos é feita por condições de falha e de sucesso, através dos lugares de entrada e de saída.

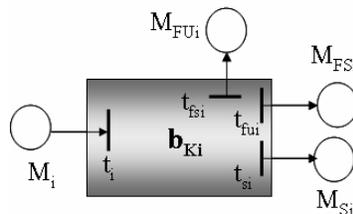


Figura 4.7 Bloco Intermediário

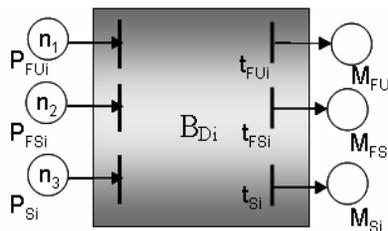


Figura 4.8 Bloco intermediário de decisão

Os blocos de um diagrama intermediário, em geral conectados em série, paralelo ou em uma configuração m/n , apresentam uma condição de falha quando:

- **conexão serial:** pelo menos um dos blocos estiver em falha;
- **conexão paralela:** todos os blocos estiverem em falha;
- **conexão m/n :** quando pelo menos m de um total de n blocos estiverem em falha.

A forma de composição de um diagrama de blocos intermediários, por meio de blocos intermediários individuais, tem na conexão m/n seu caso mais geral, uma vez que as conexões serial e paralela podem ser consideradas casos especiais quando $m=1$ e $m=n$, respectivamente. Essas formas de composição, além de permitirem a geração das configurações dos diagramas de sistema por meios gráficos, permitem ainda a obtenção de expressões de dependabilidade, em especial, de confiabilidade e disponibilidade. Para que estas composições entre blocos sejam validadas, é necessário que haja uma relação formal entre as composições dos diagramas de blocos intermediários e as expressões analíticas para confiabilidade e disponibilidade associadas. Essa formalização é obtida por meio das estatísticas de ordem [140].

4.4.3.1 Métodos de composição

Para que seja possível a avaliação de sistemas complexos, é necessário que sejam definidas regras de composição para os blocos que constituem o sistema. Duas configurações são consideradas básicas para a configuração de qualquer sistema, as configurações serial e paralela. As configurações não perfeitamente seriais, nem perfeitamente paralelas, poderão ser obtidas por meio dos métodos de composição envolvendo blocos intermediários e blocos de decisão intermediários. Métodos que possibilitem a análise devem evitar a explosão de estados, além de preservar as propriedades do sistema como *liveness*, limitação e reversibilidade. Uma das maneiras utilizadas, por exemplo, nos procesos de manufatura, sugere a adoção de soluções de síntese para modelagem de sistemas por meio dos métodos *bottom-up*, *top-down* e híbridos [45], que satisfaça as condições acima definidas. Considerando-se os blocos intermediários como sendo representações agregadas em alto nível das redes de Petri ou blocos bem formados, a serem definidos na Seção 4.4.4, podem-se utilizar os conceitos de fusão e de junção na descrição de sua composição. As possibilidades de composição entre os blocos intermediários seguem as regras de redução simples e os métodos de composição modular, ou síntese *bottom-up*, das redes de Petri. O termo fusão nesta Tese é definido como a sincronização dos blocos intermediários com a correspondente fusão dos lugares de saída, enquanto o termo junção corresponde a fusão de lugares em série, conforme definido nas regras de redução simples das redes de Petri [105][128], vistas nas seções 2.3.2 e 2.3.3 do Capítulo 2.

As possibilidades de composição, entre os blocos intermediários e os blocos de decisão intermediários, definem um conjunto de composições básicas a serem utilizadas no diagrama de blocos intermediários. As seguintes composições são possíveis:

- composição serial entre blocos intermediários;
- composição paralela entre blocos intermediários;
- composição de dois blocos intermediários concorrentes em série com bloco de decisão intermediário;
- composição entre um bloco intermediário e dois blocos intermediários concorrentes.

• **Composição Serial entre Blocos Intermediários:**

A composição de dois blocos intermediários, conforme observado na Figura 4.9a, feita por meio de fusão e/ou junção de lugares, é apresentada na Figura 4.9b. O bloco b_{ki} pode ser um bloco equivalente, o qual é o resultado da combinação de blocos intermediários em paralelo com um bloco de decisão intermediário, um bloco intermediário ou um bloco de decisão intermediário. Considerando-se inicialmente b_{ki} como sendo um bloco intermediário, embora as mesmas considerações possam ser feitas para as demais situações, pode-se definir a composição de blocos em série pelas seguintes condições:

- a) Junção do lugar de saída P_{Si} , que representa a execução do bloco b_{ki} com sucesso, com o lugar de entrada P_j do próximo bloco b_{kj} , formando o lugar único P_{Sij} , de
- b) modo a prover um caminho confiável entre os lugares P_i , entrada do primeiro bloco b_{ki} , com o lugar P_{Sj} , saída do segundo bloco b_{kj} , conforme a Figura 4.9b;
- c) Fusão do lugar de saída P_{FU_i} do bloco b_{ki} com o lugar P_{FU_j} do bloco b_{kj} formando o lugar P_{FU_i, FU_j} do bloco composto;
- d) Fusão do lugar de saída P_{FS_i} do bloco b_{ki} com o lugar P_{FS_j} do bloco b_{kj} formando o lugar P_{FS_i, FS_j} do bloco composto;
- e) Adição das transições $t_{FU_{ij}}$ e $t_{FS_{ij}}$, e dos lugares de saída do bloco composto $P_{FU_{ij}}$ e $P_{FS_{ij}}$, correspondente a falha não detectada e falha detectada, respectivamente.

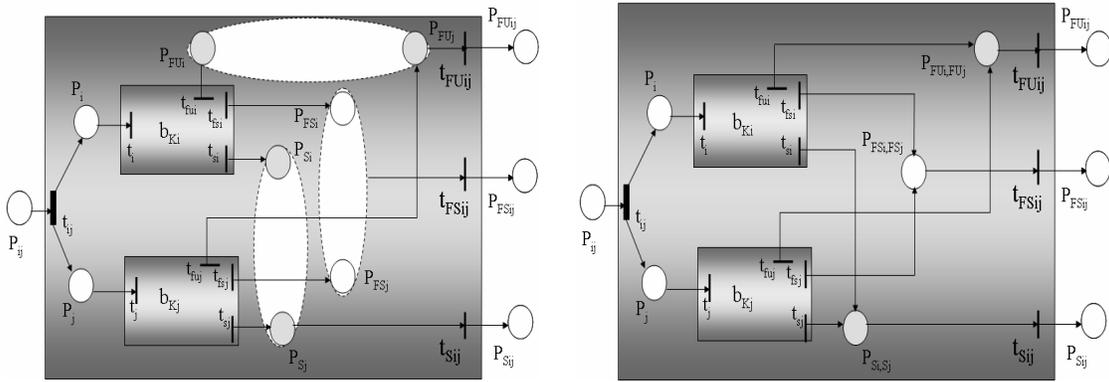
Formalmente, este processo de junção e fusão é descrito, considerando-se que o diagrama de blocos intermediários do sistema é composto por apenas dois blocos intermediários em série. Considere-se a estrutura da rede de Petri $N=(P,T,I,O)$ formada apenas pelos seguintes elementos:

- P - conjunto de lugares de entrada e de saída dos dois blocos intermediários;
- T - conjunto de transições de entrada e de saída dos dois blocos intermediários;
- I - conjunto de arcos de entrada dos dois blocos intermediários;
- O - conjunto de arcos de saída dos dois blocos intermediários.

Observando-se a Figura 4.9a, tem-se que o conjunto $\{P_{Si}, P_j\}$ representa os lugares a serem juntados; os conjuntos $\{P_{FU_i}, P_{FU_j}\}$ e $\{P_{FS_i}, P_{FS_j}\}$ representam os lugares a serem fundidos e o conjunto $\{P_{FU_{ij}}, P_{FS_{ij}}\}$ representa os lugares de saída a serem adicionados. Considerando-se ainda que P_m represente o conjunto de todos os lugares fundidos e

- c) Fusão do lugar de saída P_{FSi} do bloco b_{ki} com o lugar de saída P_{FSj} do bloco b_{kj} , correspondentes à execução com falha detectada dos blocos, formando o lugar de saída $P_{FSi,FSj}$ no bloco composto;
- d) Adição das transições t_{FUij} , t_{FSij} , t_{Sij} e t_{ij} , do lugar de entrada P_{ij} , e dos lugares de saída do bloco composto P_{Sij} , P_{FUij} e P_{FSij} , correspondente à execução com sucesso, com falha não detectada e com falha detectada, respectivamente.

A composição dos blocos intermediários paralelos segue uma descrição formal que envolve lugares, transições e arcos de uma rede de Petri $N=(P,T,I,O)$. Considerando-se que o diagrama intermediário é composto por dois blocos em paralelo, tem-se que $P_m=\{P_{Si},P_{Sj}\}\cup\{P_{FUi},P_{FUj}\}\cup\{P_{FSi},P_{FSj}\}$ representa o conjunto de lugares a serem fundidos, onde $P_m \subseteq P$, e $P_a=\{P_{FUij}, P_{FSij}, P_{Sij}$ e $P_{ij}\}$ e $T_a=\{t_{ij},t_{Sij},t_{FUij}, t_{FSij}\}$ representam o conjunto de lugares e transições a serem adicionadas ao processo, de modo a transformar a estrutura da rede de Petri $N=(P,T,I,O)$ em uma nova estrutura de rede de Petri $N'=(P',T',I',O')$.



a) Composição de blocos em paralelo b) Bloco paralelo composto equivalente

Figura 4.10 Processo de composição de blocos intermediários concorrentes

Os elementos da nova rede de Petri são definidos como:

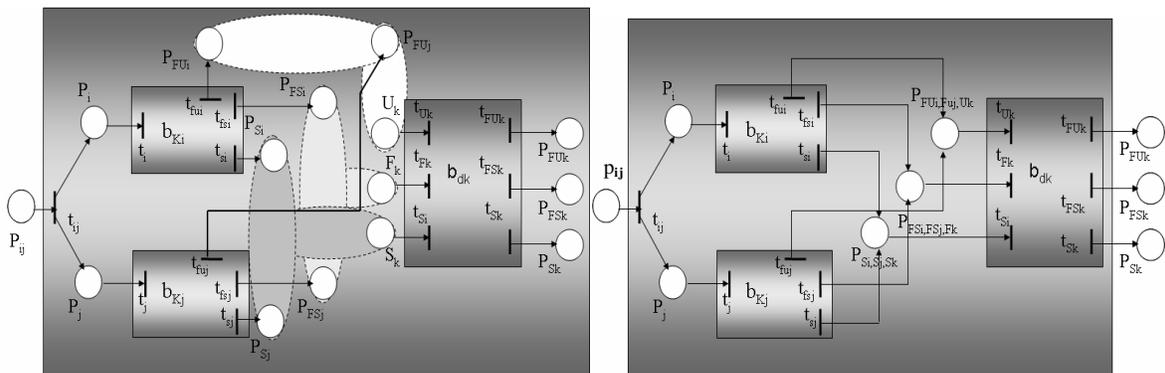
1. $T' = T \cup T_a$;
2. $P' = ((P - P_m) \cup \{P_{Si,Sj}, P_{FSi,FSj}, P_{FUi,FUj}\} \cup P_a)$, onde $(P_{Si,Sj}, P_{FSi,FSj}, P_{FUi,FUj}) \in P$ e $(P_{FUij}, P_{FSij}, P_{Sij} \text{ e } P_{ij}) \notin P$, sendo $P_{Si,Sj}$ o resultado da fusão dos lugares P_{Si} com P_{Sj} , $P_{FSi,FSj}$ o resultado da fusão dos lugares P_{FSi} com P_{FSj} e $P_{FUi,FUj}$ o resultado da fusão dos lugares P_{FUi} com P_{FUj} . Os elementos P_{FUij} , P_{FSij} e P_{Sij} representam lugares adicionais da rede de Petri $N'(P',T',I',O')$ correspondentes às saídas de falha não detectada, de falha detectada e de sucesso, respectivamente, e P_{ij} representa o lugar de entrada do bloco composto;
3. $I' = I \cup \{(P_{FUi,FUj}, t_{FUij}), (P_{FSi,FSj}, t_{FSij}), (P_{Si,Sj}, t_{Sij}), (P_{ij}, t_{ij})\}$;
4. $O' = O - \{(P_{Si}, t_{si}), (P_{Sj}, t_{sj}), (P_{FSi}, t_{fsi}), (P_{FSj}, t_{fsj}), (P_{FUi}, t_{fui}), (P_{FUj}, t_{fuj})\} \cup \{(P_{Si,Sj}, t_{si}), (P_{Si,Sj}, t_{sj}), (P_{FSi,FSj}, t_{fsi}), (P_{FSi,FSj}, t_{fsj}), (P_{FUi,FUj}, t_{fui}), (P_{FUi,FUj}, t_{fuj}), (P_{ij}, t_{ij}), (P_{ij}, t_{ij}), (P_{FSij}, t_{FSij}), (P_{FUij}, t_{FUij}) \text{ e } (P_{Sij}, t_{Sij})\}$ (4.2)

- **Composição de dois blocos intermediários concorrentes em série com bloco de decisão intermediário**

A composição de blocos intermediários com blocos de decisão, observado na Figura 4.11a, e executada por meio de fusão e/ou junção de lugares, é apresentada na Figura 4.11b. As conexões paralelas, entre blocos intermediários terminais, são sempre finalizadas em um bloco de decisão. Pode-se definir a composição dos blocos paralelos com o bloco de decisão pelas seguintes condições:

- Fusão do lugar de saída P_{FU_i} , do bloco intermediário b_{k_i} , com o lugar de saída P_{FU_j} , do bloco intermediário b_{k_j} , correspondentes à execução com falha não detectada dos blocos intermediários, formando o lugar de saída P_{FU_i, FU_j} ;
- Fusão do lugar de saída P_{FS_i} do bloco intermediário b_{k_i} com o lugar de saída P_{FS_j} do bloco intermediário b_{k_j} , correspondentes à execução com falha detectada dos blocos intermediários, formando o lugar de saída P_{FS_i, FS_j} ;
- Fusão do lugar de saída P_{S_i} , do bloco b_{k_i} , com o lugar de saída P_{S_j} , do bloco b_{k_j} , correspondentes à execução com sucesso dos blocos intermediários, formando o lugar P_{S_i, S_j} ;
- Junção do lugar de entrada U_k , do bloco de decisão intermediário b_{dk} , com o lugar de saída P_{FU_i, FU_j} , formando o lugar P_{FU_i, FU_j, U_k} ;
- Junção do lugar de entrada F_k , do bloco de decisão intermediário b_{dk} , com o lugar de saída P_{FS_i, FS_j} , formando o lugar P_{FU_i, FU_j, F_k} ;
- Junção do lugar de entrada S_k , do bloco de decisão intermediário b_{dk} , com o lugar de saída P_{S_i, S_j} , formando o lugar de saída P_{S_i, S_j, S_k} ;
- Adição da transição t_{ij} e do lugar de entrada P_{ij} .

Considerando-se que o diagrama intermediário é composto por dois blocos intermediários em paralelo, em série com um bloco de decisão intermediário, tem-se que $P_m = \{P_{S_i}, P_{S_j}, S_k\} \cup \{P_{FU_i}, P_{FU_j}, U_k\} \cup \{P_{FS_i}, P_{FS_j}, F_k\}$ representa o conjunto de lugares de saída e de entrada, que passaram por fusão seguido por junção, onde $P_m \subseteq P$, e $P_a = \{P_{ij}\}$ e $T_a = \{t_{ij}\}$ representa o lugar e a transição de entrada a serem adicionados ao processo, de modo a transformar a estrutura da rede de Petri $N = (P, T, I, O)$, em uma nova estrutura de rede de Petri $N' = (P', T', I', O')$.



a) Processo de composição paralelo/serial

b) Bloco composto equivalente

Figura 4.11 Processo de composição de blocos intermediários com bloco de decisão

Os elementos da nova estrutura de rede de Petri são definidos como:

1. $T' = T \cup T_a$;
2. $P' = (P - P_m) \cup \{P_{S_i, S_j, S_k}, P_{F_{S_i, F_{S_j}, F_k}}, P_{F_{U_i, F_{U_j}, U_k}}\} \cup P_a$, onde $(P_{S_i, S_j, S_k}, P_{F_{S_i, F_{S_j}, F_k}}, P_{F_{U_i, F_{U_j}, U_k}}, P_{ij}) \notin P$, sendo $(P_{S_i, S_j, S_k}, P_{F_{S_i, F_{S_j}, F_k}}, P_{F_{U_i, F_{U_j}, U_k}})$ o resultado obtido das fusões e junções das condições anteriores e P_{ij} representa o lugar de entrada do bloco composto;
3. $I' = I \cup \{(P_{ij}, t_{ij})\}$;
4. $O' = O \cup \{(P_i, t_{ij}), (P_j, t_{ij})\}$. (4.3)

- **Composição entre um bloco intermediário e dois blocos intermediários concorrentes:**

A composição de um bloco intermediário em série com dois blocos intermediários em paralelo é processada em duas etapas:

a) na primeira etapa, os dois blocos intermediários em paralelo são transformados em um bloco paralelo composto equivalente, conforme descrito anteriormente e mostrado na Figura 4.10b;

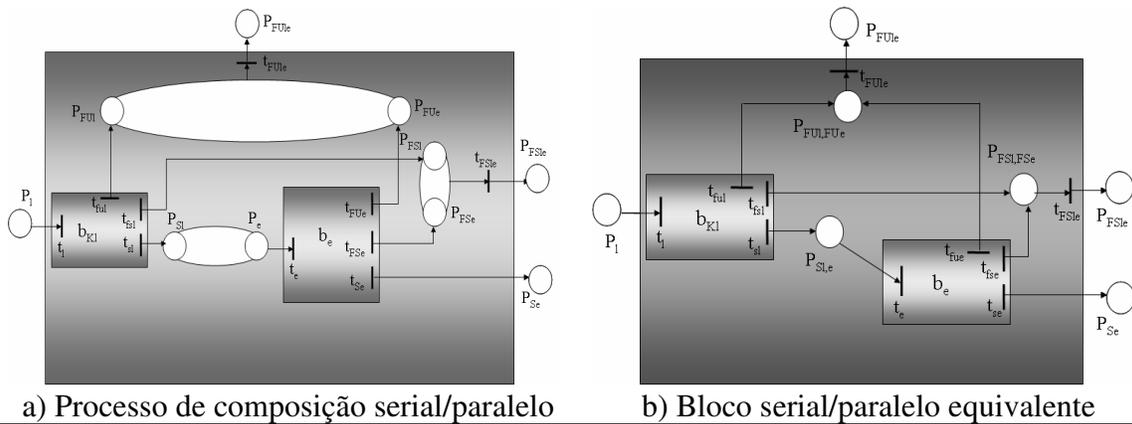
b) na segunda etapa processa-se a composição serial do bloco intermediário com o bloco paralelo composto equivalente, conforme pode ser observado na Figura 4.12a.

Na Figura 4.12b, observa-se a composição do bloco intermediário em série com o bloco paralelo composto equivalente por meio de junção e fusões. Pode-se definir este processo de composição por intermédio das seguintes condições:

- a) **Junção** do lugar de saída P_{S_i} , que representa a execução com sucesso do bloco b_{kl} , com o lugar de entrada P_e do bloco composto equivalente b_e , formando o lugar único $P_{S_{ie}}$, de modo a prover um caminho confiável entre os lugares de entrada P_i , do primeiro bloco b_{kl} , com o lugar de saída P_{S_e} , do bloco composto equivalente b_e , conforme a Figura 4.12a;
- b) **Fusão** do lugar de saída $P_{F_{U_i}}$, do bloco b_{kl} , com o lugar de saída $P_{F_{U_e}}$, do bloco composto equivalente b_e , formando o lugar $P_{F_{U_i, F_{U_e}}}$ do bloco composto;
- c) **Fusão** do lugar de saída $P_{F_{S_i}}$, do bloco b_{kl} , com o lugar de saída $P_{F_{S_e}}$, do bloco composto equivalente b_e , formando o lugar $P_{F_{S_i, F_{S_e}}}$ do bloco composto;
- d) **Adição** das transições $t_{F_{U_e}}$ e $t_{F_{S_{ie}}}$, e dos lugares de saída $P_{F_{U_e}}$ e $P_{F_{S_{ie}}}$ do bloco composto equivalente b_e , correspondente à falha não detectada e à falha detectada, respectivamente.

A descrição formal desse modelo é executada em dois passos, de acordo com as considerações anteriormente expostas:

- 1) O primeiro passo é aquele correspondente à descrição formal entre blocos paralelos em série com bloco de decisão, conforme as Figuras 4.11a e 4.11b;



a) Processo de composição serial/paralelo

b) Bloco serial/paralelo equivalente

Figura 4.12 Processo de composição de blocos intermediários com bloco de decisão

2) O segundo passo é aquele correspondente à descrição formal entre dois blocos em série, conforme as Figuras 4.9a e 4.9b. Neste caso, porém, o segundo bloco em série corresponde ao bloco composto equivalente obtido no 1º passo.

Este processo de fusão e junção entre blocos intermediários estende-se para todos os blocos adjacentes em série. Ao final das etapas de síntese, o diagrama de blocos intermediários, que representa o sistema como um todo, é obtido, conforme pode ser visto na Figura 4.13. O diagrama de blocos intermediários mostrado na Figura 4.13 representa o diagrama de blocos estendido em dependabilidade da Figura 4.6. O diagrama de blocos intermediário descreve as conexões entre os blocos que compõem o diagrama, conforme regras de composição formais anteriormente definidas.

Considerando-se o diagrama da Figura 4.6, observa-se que este diagrama possui conexões seriais e paralelas bem definidas, o que permite a aplicação direta de fórmulas analíticas para análise de confiabilidade e disponibilidade por meio, por exemplo, das estatísticas de ordem [140]. Caso as conexões não sejam exatamente seriais ou paralelas, pode-se resolver o diagrama por meio da aplicação do teorema das probabilidades totais de Bayes [136], ou por aproximações por limites superior e inferior [126].

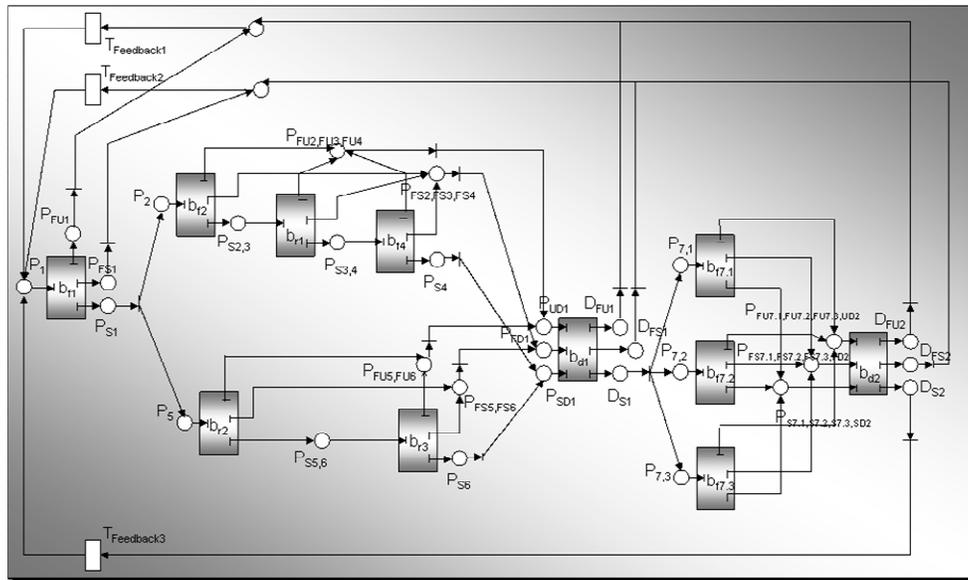


Figura 4.13 Diagrama de blocos intermediário correspondente ao diagrama EDBD

Uma forma alternativa e prática é a utilização da presente metodologia para transformação desses diagramas, que não seguem exatamente o modelo serial/paralelo, inicialmente em um diagrama de blocos intermediários, conforme descrito anteriormente, e, em seguida, em um diagrama dependável, para obtenção dos valores de dependabilidade, conforme será visto em exemplo mais adiante.

Os blocos intermediários, conforme definidos no nível hierárquico 3, representam a configuração dos blocos de um diagrama EDBD por meio de redes de Petri de alto nível. A representação do comportamento de cada bloco do diagrama EDBD, por meio do modelo EDSPN correspondente, será executada no nível hierárquico 4.

4.4.4 Nível Hierárquico 4: Modelos EDSPN

No nível hierárquico 4, procede-se à análise de cada bloco, isoladamente, de modo a se obter um maior detalhamento de suas características. Quanto mais preciso for o modelo, maior deverá ser o nível de detalhes incorporados ao mesmo. Através da metodologia *top-down*, dois esquemas de refinamentos são passíveis de utilização: expansão de lugares e expansão de transições [45]. Aplicações do método *top-down* são limitadas a sistemas formados por blocos independentes, ou seja, blocos cujos recursos sejam compartilhados apenas internamente. O método *top-down*, inicialmente proposto por Valette [142], assume que, caso o disparo de uma transição não seja instantâneo, porém feito em duas etapas, então a associação de uma transição a uma operação complexa é possível, a qual pode ser descrita por meio de uma outra rede de Petri.

A Figura 4.14 descreve o que se denomina de bloco agregado ou bloco bem formado. Este tipo de bloco é definido como uma rede de Petri, constituída por uma única transição de entrada denominada transição inicial T_i , e uma única transição de saída, denominada transição final T_f . A rede de Petri associada a um bloco bem formado, que deverá

substituir a transição T_i da Figura 4.14, é definida como a rede que resulta da adição de um lugar P_0 ao bloco, denominado *idle place*, desde que a transição de saída do lugar P_0 corresponda à transição inicial do bloco e a transição de entrada do lugar P_0 corresponda à transição final do bloco. De modo a preservarem-se importantes propriedades qualitativas das redes de Petri, tais como ausência de *deadlock*, segurança e limitação, após a troca da transição do T_i do bloco bem formado por uma rede de Petri complexa, as seguintes condições deverão ser obedecidas [45]:

- a rede de Petri associada deve ser viva, i.e., não ter *deadlock*;
- a marcação inicial da rede de Petri associada é a única marcação em que o lugar P_0 , *idle place*, não está vazio, i.e., sem *token*.
- a única transição habilitada pela marcação inicial é a transição inicial.

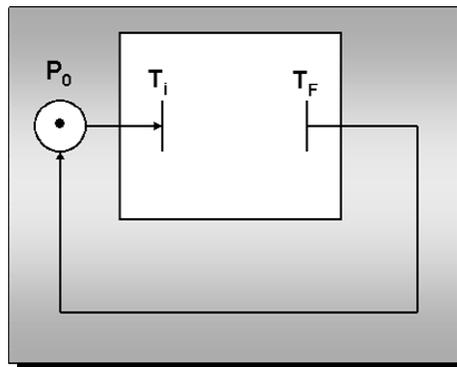


Figura 4.14 Bloco bem formado

Os blocos intermediários correspondem à definição de blocos bem-formados de Valette [142]. Os blocos intermediários possuem três possíveis saídas, ao invés de apenas uma, como nos blocos bem-formados. Contudo, como as três saídas são exclusivas, elas podem ser consideradas como sendo uma única saída, o que valida a definição dos blocos bem-formados. Portanto, pode-se trocar a transição inicial T_i de cada bloco intermediário por uma rede de Petri associada, correspondente ao modelo MDP, a ser definido no nível hierárquico 5, para cada bloco do diagrama EDBD. Contudo, para que os modelos MDP do nível hierárquico 5 possam ser analisados, é necessário um maior detalhamento dos atributos de confiabilidade, disponibilidade e segurança relacionados a cada bloco do diagrama EDBD, isoladamente. Pode-se obter esse detalhamento por meios analíticos ou numéricos, através dos modelos EDSPN. Os modelos EDSPN, nesta metodologia, representam os comportamentos dos blocos de um diagrama EDBD com relação aos atributos de dependabilidade, sejam eles valores numéricos ou expressões analíticas.

Blocos cujas distribuições de falha sejam não Markovianas, porém compostas por distribuições de falhas exponenciais como, por exemplo, as distribuições do tipo Erlang, hipoeponencial e hiperexponencial, são analisadas pelo método de fases [139]. Nas distribuições hipoeponenciais e Erlang as fases são seqüenciais, enquanto nas distribuições hiperexponenciais as fases são alternadas [101]. Outras técnicas de análises das distribuições não-Markovianas são as variáveis suplementares [60], a teoria da renovação de Markov [139] e simulação de eventos discretos [31]. Blocos cujas distribuições estatísticas sejam não-Markovianas e que não possam ser avaliadas como

um conjunto de exponenciais serão objeto de avaliações específicas. Estas avaliações específicas podem estar relacionadas a obtenção das expressões analíticas dos atributos de dependabilidade, ou ainda à obtenção de valores numéricos por meio de análise ou simulação dos modelos EDSPN. Esses valores numéricos podem ser obtidos através de avaliação transiente ou de estado permanente do modelo EDSPN, por meio de ferramentas de modelagem do tipo *TimeNet* [59][134]. Os valores numéricos ou expressões analíticas de dependabilidade obtidas para cada bloco são transferidos para as transições imediatas dos modelos MDP no nível hierárquico 5. A seguir, são descritos alguns exemplos de modelos EDSPN.

4.4.4.1 Exemplo I: Modelo EDSPN sem Cobertura de Falhas

Neste modelo, distribuições associadas a eventos de falha e reparo são apresentados na forma de distribuições exponenciais, o que possibilita a análise de disponibilidade por meio de modelos Markovianos, conforme observado na Figura 4.15. Aspectos relacionados às estimativas de segurança não serão considerados, o que deverá ser feito no próximo exemplo.

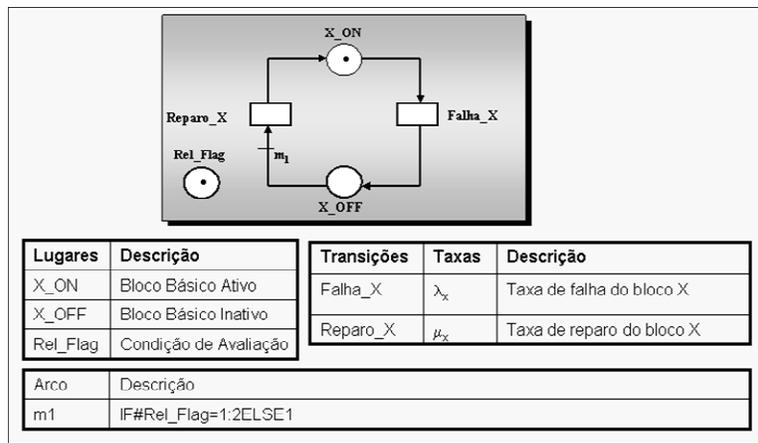


Figura 4.15 Modelo EDSPN sem cobertura de falhas para o bloco básico X

No modelo EDSPN da Figura 4.15, os parâmetros MTTF e MTTR do bloco funcional X, cujas distribuições de falha e de reparo são distribuições exponenciais, são representados pelos retardos das transições Falha_X e Reparo_X, respectivamente. As duas possíveis marcações (estados) deste modelo correspondem à presença de *token* nos lugares X_ON e X_OFF e a alternância entre eles depende das respectivas taxas de falha $\lambda_x = (1/MTTF)$ e de reparo $\mu_x = (1/MTTR)$. Considerando-se que, a expressão analítica de confiabilidade do bloco X segue uma distribuição exponencial, obtida por meio de uma distribuição geral de Poisson [126] e que a taxa de falha λ_x é constante durante um intervalo de tempo t, a expressão da confiabilidade é dada pela expressão $R(t) = e^{-\lambda_x t}$.

Considerando-se que o bloco X é reparável e que a taxa de falha λ_x e a taxa de reparo μ_x sejam constantes, a expressão de disponibilidade de estado permanente é expressa por:

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{\frac{1}{\lambda_x}}{\frac{1}{\lambda_x} + \frac{1}{\mu_x}} = \frac{\mu_x}{\lambda_x + \mu_x} = \frac{1}{1 + \left(\frac{\lambda_x}{\mu_x}\right)} \quad (4.4)$$

A expressão de disponibilidade pode ser também obtida, por exemplo, analisando-se a CTMC por meio das cadeias de nascimento e morte [140][22] e das equações de balanceamento.

As expressões analíticas de confiabilidade e de disponibilidade, para o modelo EDSPN correspondente ao bloco X, além das métricas utilizadas para a sua obtenção são mostradas na Tabela 4.1.

Tabela 4.1 Expressões analíticas e métricas do modelo EDSPN sem cobertura de falhas

Atributo	Tipo de Avaliação	Expressão analítica	Métricas
Confiabilidade	Transiente	$R(t) = \exp^{-\lambda_x t}$	$P\{\#X_ON=1\}$
Inconfiabilidade	Transiente	$Q(t) = 1 - \exp^{-\lambda_x t}$	$P\{\#X_ON=0\}$
Disponibilidade	Estado Permanente	$A = \frac{1}{1 + \left(\frac{\lambda_x}{\mu_x}\right)}$	$P\{\#X_ON=1\}$
Indisponibilidade	Estado Permanente	$U = 1 - \frac{1}{1 + \left(\frac{\lambda_x}{\mu_x}\right)}$	$P\{\#X_ON=0\}$

As métricas de confiabilidade e de disponibilidade, apesar de apresentarem a mesma expressão na ferramenta de modelagem, têm interpretações diferentes:

- Métrica de confiabilidade: Define a probabilidade do bloco encontrar-se ativo (operacional) num intervalo de tempo variando do tempo zero ao tempo t, dado que o bloco se encontra perfeito no instante de tempo zero. A métrica de confiabilidade corresponde à expressão $P\{\#X_ON=1\}$, a qual indica a presença de um *token* no lugar X_ON para um dado intervalo de tempo;
- Métrica de disponibilidade: Define a probabilidade que o bloco esteja operacional (ativo) em um dado instante de tempo. A métrica de disponibilidade corresponde à expressão $P\{\#X_ON=1\}$, a qual indica a presença de um *token* no lugar X_ON num dado instante de tempo.

4.4.4.2 Exemplo II: Modelo EDSPN com Cobertura de Falhas

Este modelo, mostrado na Figura 4.16, um pouco mais complexo, envolve expressões numéricas ou analíticas para confiabilidade, disponibilidade e segurança. Neste modelo, o tempo associado à falha é exponencialmente distribuído com taxa

constante λ_x , e os tempos de reparo, também exponencialmente distribuídos, apresentam duas taxas constantes e distintas, μ_{x1} e μ_{x2} . A taxa μ_{x1} corresponde ao tempo de reparo propriamente dito, considerando-se que a falha seja detectada quando de sua ocorrência. A taxa μ_{x2} corresponde à soma do tempo de reparo mais o tempo de percepção do erro. A probabilidade de detecção do erro, quando da ocorrência da falha, é representada pelo fator de cobertura C_x . Neste caso, como a taxa de reparo pode assumir valores distintos em fases alternadas, ou seja, μ_{x1} para um fator de cobertura C_x , e μ_{x2} para um fator de cobertura $1-C_x$, observa-se que o processo de reparo apresenta uma distribuição hiperexponencial.

No modelo EDSPN da Figura 4.16, os parâmetros MTTF, MTTR e MTEP são representados pelos retardos das transições Falha_X, Reparo_X e Percep_X, respectivamente, para um bloco funcional X, cujas distribuições de falha e de reparo são distribuições exponenciais. As três possíveis marcações (estados), deste modelo, correspondem à presença de tokens nos lugares X_ON, X_OFF_Det e X_OFF_Não_Det, cuja alternância entre eles depende das respectivas taxas de falha $\lambda_x=(1/MTTF)$ e de reparo $\mu_{x1}=(1/MTTR)$ e $\mu_{x2}=(1/MTEP)$. A expressão analítica de confiabilidade do bloco X com cobertura de falhas é dada por $R(t) = e^{-\lambda t}$, a mesma do bloco X sem cobertura de falhas, uma vez que a análise transiente termina quando o bloco falha, isto é, quando o lugar X_OFF recebe o *token* que estava em X_ON.

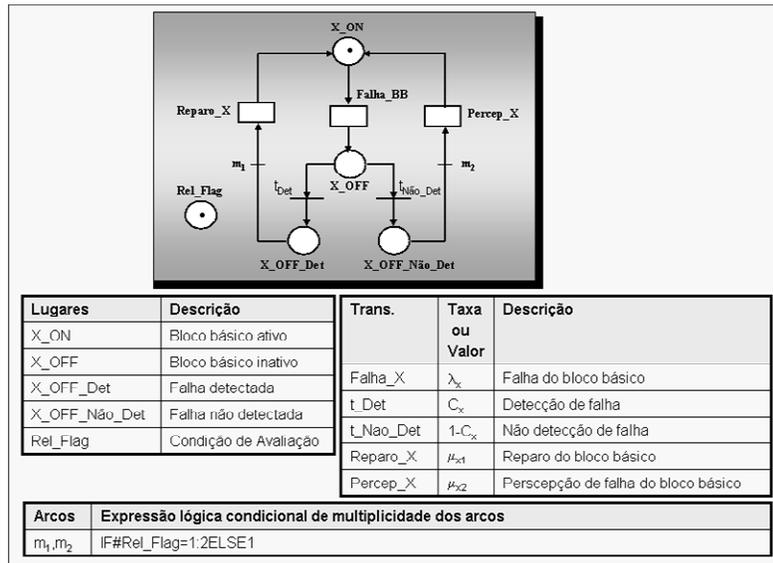


Figura 4.16 Modelo EDSPN com cobertura de falhas para o bloco X.

A probabilidade de ocorrência de uma das fases, definida por C_x , resulta da detecção da falha, enquanto que a probabilidade de ocorrência da outra fase, definida por $(1- C_x)$, resulta da não detecção da falha ou da detecção da falha, após decorridos um longo tempo de sua ocorrência. Dependendo da aplicação, falhas não detectadas têm efeitos catastróficos.

Considerando-se que o bloco X é reparável e que a taxa de falha λ_x e as taxas de reparo μ_{x1} e μ_{x2} sejam constantes, a expressão de disponibilidade de estado permanente é expressa por:

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTTF + (MTTR_1 + MTTR_2)} = \frac{\frac{1}{\lambda_x}}{\frac{1}{\lambda_x} + \left(\frac{C_x}{\mu_{x1}} + \frac{(1-C_x)}{\mu_{x2}} \right)}$$

$$A = \frac{1}{1 + \lambda_x \left(\frac{C_x}{\mu_{x1}} + \frac{(1-C_x)}{\mu_{x2}} \right)} \quad (4.5)$$

Conforme pode ser observado em [136][140], a expressão analítica da disponibilidade de estado permanente depende somente dos tempos médios de falha (MTTF) e de reparo (MTTR), e não da natureza das distribuições dos tempos de falha e de reparo. Uma outra maneira de se obter esta expressão seria através da análise da CTMC, por meio das cadeias de nascimento e morte e das equações de balanceamento de fluxo [140], conforme mostrado no exemplo anterior.

Para o bloco básico, as estimativas de confiabilidade e de disponibilidade, além das métricas utilizadas para a sua obtenção, são mostradas na Tabela 4.2.

Tabela 4.2 Expressões analíticas e métricas do modelo EDSPN com cobertura de falhas

Atributo	Tipo de Avaliação	Expressão analítica	Métricas
Confiabilidade	Transiente	$R(t) = \exp^{-\lambda_x t}$	$P\{\#X_{ON}=1\}$
Inconfiabilidade	Transiente	$Q(t) = 1 - \exp^{-\lambda_x t}$	$P\{\#X_{ON}=0\}$
Falha Segura	Transiente	$FS(t) = (1 - \exp^{-\lambda_x t})C_x$	$P\{\#X_{OFF_Det}=1\}$
Falha Insegura	Transiente	$FU(t) = (1 - \exp^{-\lambda_x t})(1 - C_x)$	$P\{\#X_{OFF_Não_Det}=1\}$
Segurança	Transiente	$S(t) = \exp^{-\lambda_x t} + (1 - \exp^{-\lambda_x t})C_x$	$P\{\#X_{ON}+\#X_{OFF_Det}=1\}$
Disponibilidade	Estado Permanente	$A = \frac{1}{1 + \lambda_x \left(\frac{C_x}{\mu_{x1}} + \frac{(1-C_x)}{\mu_{x2}} \right)}$	$P\{\#X_{ON}=1\}$
Indisponibilidade	Estado Permanente	$U = 1 - \frac{1}{1 + \lambda_x \left(\frac{C_x}{\mu_{x1}} + \frac{(1-C_x)}{\mu_{x2}} \right)}$	$P\{\#X_{ON}=0\}$

As métricas de confiabilidade, falha segura, falha insegura, segurança e disponibilidade podem ser descritas da forma seguinte:

a) métrica de confiabilidade: definida pela expressão $P\{\#X_ON=1\}$, corresponde a presença de um *token* no lugar X_ON , indica a condição operacional do bloco X para um dado intervalo de tempo;

b) métrica de falha segura: definida pela expressão $P\{\#X_OFF_Det=1\}$, corresponde a presença de um *token* no lugar X_OFF_Det , indica a condição de falha do bloco X para um dado intervalo de tempo, dado que a ocorrência da falha tenha sido detectada;

c) métrica de falha insegura: definida pela expressão $P\{\#X_OFF_Não_Det=1\}$, corresponde a presença de um *token* no lugar $X_OFF_Não_Det$, indica a condição de falha do bloco X para um dado intervalo de tempo, porém sem que a ocorrência da falha tenha sido detectada. A métrica de falha insegura ou métrica de insegurança define a probabilidade de ocorrência de falhas catastróficas;

d) métrica de segurança: definida pela expressão $P\{\#X_ON+\#X_OFF_Det=1\}$, corresponde a presença de um *token* no lugar $\#X_ON$ ou no lugar X_OFF_Det , indica a condição operacional do bloco X para um dado intervalo de tempo, ou mesmo uma condição de falha, porém detectável;

e) métrica de disponibilidade: definida pela expressão $P\{\#X_ON=1\}$, corresponde à presença de um *token* no lugar X_ON , indica a condição de operacionalidade do bloco X em um dado instante de tempo.

Caso o bloco seja formado por dispositivos de hardware e de software, a definição de uma expressão analítica, que leve em conta o funcionamento conjunto dos componentes, pode não ser uma tarefa simples. Isto conduz o modelador a uma nova estratégia de definição da função de distribuição de probabilidade do modelo, conforme descrição a seguir.

4.4.4.3 Exemplo III: Modelo EDSPN composto por hardware e software

Neste modelo EDSPN, considera-se que o bloco funcional é constituído por componentes de hardware e de software. As distribuições associadas a eventos de falha e de reparo, para ambos os componentes, são distribuições exponenciais, conforme pode ser observado na Figura 4.17.

Os lugares e as transições do modelo EDSPN, correspondentes ao bloco, composto por hardware e software, são mostrados na Figura 4.17. A multiplicidade dos arcos é definida por expressões lógicas condicionais em função das marcações na Tabela 4.3, e as habilitações dos disparos das transições imediatas são definidas por meio de expressões de guarda, em função das marcações na Tabela 4.4. As condições de dependência entre os componentes de hardware e software, em um determinado bloco, podem ocorrer dos seguintes modos:

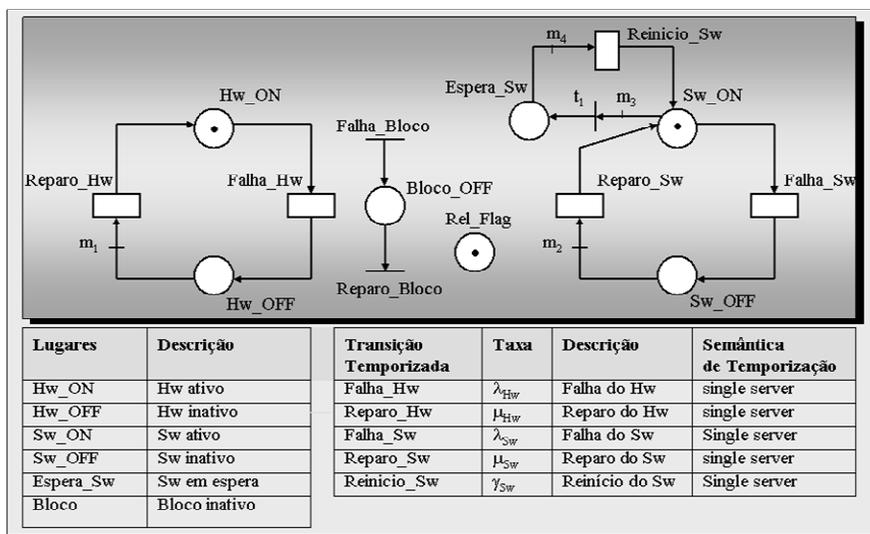


Figura 4.17 Modelo EDSPN para blocos compostos por Hw e Sw

- Quando o hardware falha, a execução do software é paralisada e este vai a uma condição de espera. O software continua em uma condição perfeita de funcionamento, porém na espera do reparo do hardware. Quando o hardware é recuperado, o software é reiniciado e o bloco torna-se operacional de novo;
- Quando o software falha, o hardware continua operacional. Quando o software é restaurado, a execução do bloco é reiniciada;
- Quando o hardware e o software encontram-se em falha, caso o software seja restaurado antes do hardware, o software é colocado em espera e, só após a recuperação do hardware, o software é reiniciado, reiniciando, por consequência, a operação do bloco.

Tabela 4.3 Expressões de multiplicidade dos arcos do modelo EDSPN do bloco

Arco	Expressões lógicas condicionais da multiplicidade dos arcos	Descrição
m ₁	IF(#Rel_Flag=1):2ELSE1	Se análise de confiabilidade Então não permitir reparo do Hw Senão permitir reparo do Hw
m ₂	IF(#Rel_Flag=1):2ELSE1	Se análise de confiabilidade Então não permitir reparo do Sw Senão permitir reparo do Sw
m ₃	IF(#HW_ON=0):1ELSE2	Se HW falha Então SW é colocado na espera (<i>standby</i>) Senão SW permanece ativo
m ₄	IF(#Rel_Flag=1):1ELSE2	Se análise de confiabilidade Então não permitir reinício do Sw em espera Senão permitir reinício do Sw em espera

Tabela 4.4 Funções de guarda do modelo EDSPN do bloco X

Transição	Expressão dos pesos ou funções de guarda	Prioridade
Falha_Bloco	(#Hw_ON=0OR#Sw_ON=0)AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>0)AND#Bloco=1	1

As condições de dependência entre o hardware e o software no modelo EDSPN correspondente ao bloco, podem ser explicadas observando-se as ações seguintes:

- a. A falha do hardware causada pelo disparo da transição Falha_HW, retira o *token* do lugar HW_ON e o coloca no lugar HW_OFF. A falha do hardware implica na falha do bloco pela colocação de um *token* no lugar Bloco_OFF, devido ao disparo da transição Falha_Bloco. O software ao perceber a falha do hardware retira o *token* do lugar SW_ON e o coloca no lugar Espera_SW, enquanto aguarda a recuperação do Hardware. Quando o hardware se recupera, indicado pelo disparo da transição Reparo_HW, pela retirada do *token* do lugar HW_OFF e pela colocação do token no lugar HW_ON, o software é reiniciado. O reinício do software, por meio do disparo da transição Reinício_SW, retira o *token* do lugar Espera_Sw e o coloca no lugar SW_ON o que habilita o disparo da transição Reparo_Bloco, que faz com que o *token* seja retirado do lugar Bloco_OFF, fazendo o bloco retornar à condição operacional.
- b. O software falha quando o *token* é retirado do lugar SW_ON e colocado no lugar SW_OFF pelo disparo da transição Falha_SW, fazendo com que a transição Falha_Bloco dispare e um *token* seja colocado no lugar Bloco_OFF, desabilitando o bloco. O hardware, entretanto, continua operacional, isto é, com o *token* no lugar HW_ON. Quando o software é restaurado, pelo disparo da transição Reparo_SW, o *token* é retirado do lugar SW_OFF e é colocado de novo no lugar SW_ON. Isso faz com que o *token* no lugar Bloco_OFF seja retirado pelo disparo da transição Reparo_Bloco, fazendo o bloco retornar à condição operacional.
- c. Quando o hardware e o software encontram-se em falha, representado pela colocação de *token* nos lugares HW_OFF e SW_OFF devido ao disparo das transições Falha_HW e Falha_SW, o bloco é desabilitado pelo disparo da transição Falha_Bloco e a colocação de um *token* no lugar Bloco_OFF. Caso o software se recupere antes do hardware, pelo disparo da transição Reparo_SW e pela retirada do *token* do lugar SW_OFF e colocação do *token* no lugar SW_ON, este é imediatamente retirado e colocado no lugar Espera_SW, até a recuperação do hardware. Quando o hardware se recupera pelo disparo da transição Reparo_HW e pela retirada do *token* do lugar HW_OFF e colocação no lugar HW_ON, o software então retira o *token* do lugar Espera_SW pelo disparo da transição Reinício_SW e o coloca no lugar SW_ON, fazendo com que a transição Reparo_Bloco seja disparada e o *token* seja retirado do lugar Bloco_OFF. Isso faz com que o bloco retorne à sua condição de operacionalidade.

Observa-se que, a obtenção de uma expressão analítica para a confiabilidade e para a disponibilidade não é, em geral, uma tarefa simples. No caso do modelo de bloco básico,

com componentes de hardware e software, apesar das distribuições serem exponenciais, as dependências existentes entre os dois componentes dificultam a obtenção dessa expressão. Para contornar os problemas impostos pela dificuldade de obterem-se expressões analíticas de dependabilidade de um ou mais blocos, pode proceder-se à análise transiente e estacionária do modelo EDSPN correspondente a esses blocos, obter-se estimativas numéricas de dependabilidade e transferir essas estimativas para os modelos MDP no nível hierárquico 5, correspondentes aos blocos que estão sendo analisados. O modelo MDP do sistema como um todo levará em conta tanto expressões analíticas de alguns blocos, quanto valores numéricos de outros blocos obtidos para um determinado número de pontos de verificação, em intervalos de tempos regulares. Considerando-se, por exemplo, o modelo da Figura 4.17, e aplicando-se os tempos médios Falha_HW=100u.t. (onde u.t. é a abreviação de unidades de tempo), Reparo_HW=10u.t., Falha_SW=400u.t., Reparo_SW=20u.t. e Reinício_SW=1u.t. às transições correspondentes do modelo para um intervalo de tempo $t=100$ u.t., e intervalos regulares entre pontos de verificação $IT=10$ u.t., obtém-se a estimativa numérica de disponibilidade em estado permanente, conforme a Tabela 4.5 e as estimativas numéricas de confiabilidade mostradas na Tabela 4.6, para o bloco correspondente ao modelo EDSPN.

Tabela 4.5 Disponibilidade do modelo EDSPN do bloco de Hw e Sw

Atributo	Valor em %	Métricas
Disponibilidade	0.860042	$P\{\#Bloco_OFF=0\}$
Indisponibilidade	0.139958	$P\{\#Bloco_OFF=1\}$

Tabela 4.6 Confiabilidade do modelo EDSPN do bloco de Hw e Sw

Pontos de verificação	Tempo	Confiabilidade (em %)	Inconfiabilidade (em %)	Métricas
1°	10 u.t	0.8824969	0.1175031	Confiabilidade: $P\{\#Bloco_OFF=0\}$ Inconfiabilidade: $P\{\#Bloco_OFF=1\}$
2°	20 u.t	0.7788008	0.2211992	
3°	30 u.t	0.6872893	0.3127107	
4°	40 u.t	0.6065307	0.3934693	
5°	50 u.t	0.5352614	0.4647386	
6°	60 u.t	0.4723666	0.5276334	
7°	70 u.t	0.416862	0.5831380	
8°	80 u.t	0.3678794	0.6321206	
9°	90 u.t	0.3246525	0.6753475	
10°	100 u.t	0.2865048	0.7134952	

Os valores de confiabilidade e disponibilidade obtidos são então transpostos para o nível hierárquico 5, para a obtenção dos requisitos do sistema, por meio de um diagrama de modelos MDP.

Características relacionadas à multiplicidade dos arcos e às taxas de disparo dependentes da marcação serão utilizadas com bastante frequência nos modelos a serem definidos no próximo capítulo. Essas características, comuns às redes SRN, permitem a

definição de expressões lógicas condicionais [53] para as características citadas, nos modelos EDSPN, o que proporciona uma grande versatilidade e flexibilidade de modelagem, análise e refinamento.

4.4.5 Nível Hierárquico 5: Diagrama de sistema dependável e parametrizado

Os modelos EDSPN, apesar de possibilitarem a modelagem de uma grande gama de sistemas, não têm flexibilidade para representar qualquer tipo de distribuição de falha e de reparo, a menos que estas possam ser expressas como combinações de distribuições exponenciais. Como forma de contornar parte desta inflexibilidade e de reduzir os problemas devido à explosão de estados é introduzido o Modelo Dependável Parametrizado, ou simplesmente MDP. Esse modelo é formado basicamente por transições imediatas, cujos pesos são expressões numéricas ou analíticas obtidas dos modelos EDSPN no nível hierárquico 4. As expressões analíticas e/ou os valores numéricos, obtidos no nível hierárquico 4, e os respectivos parâmetros estruturais são utilizados para o cálculo dos atributos de dependabilidade de um ou mais blocos, ou de todo o sistema. Para que o modelo seja mais flexível e possa inclusive ser reusado, uma série de lugares adicionais e isolados, denominados *flags*, são definidos. As expressões ou valores dos pesos das transições imediatas em conflito e as expressões dos arcos, quando houver, são expressões lógicas dependentes da marcação.

O modelo MDP é constituído por duas camadas:

- **camada de modelo** composta pelos elementos da rede de Petri que representam o modelo MDP correspondente ao bloco do diagrama EDBD, e pelas expressões analíticas e/ou valores numéricos de confiabilidade e disponibilidade obtidos no nível hierárquico 4;
- **camada de configuração** constituída pelos *flags* que têm como função a definição da configuração ou do tipo de replicação dos elementos de um bloco, ou entre blocos distintos, e a definição do tipo e a forma das análises a serem obtidas. As configurações podem ser do tipo paralela, serial, m/n ou ainda configurações não-serial/não-paralelas [126]; os tipos de replicação podem ser passivas, semi-ativas e ativas, correspondentes às técnicas de tolerância a falhas dos blocos; as análises, quanto ao tipo, podem ser de confiabilidade ou de disponibilidade, e quanto à forma, podem ser transiente ou de estado permanente.

O modelo MDP permite que o bloco possa apresentar qualquer tipo de distribuição de confiabilidade e disponibilidade, desde que as expressões analíticas ou valores numéricos desses atributos sejam definidos e que sejam suportados pela ferramenta de modelagem. O modelo MDP de um bloco básico constituído por apenas 1 componente é mostrado na Figura 4.18. O modelo é constituído pela rede de Petri presente na camada de modelo mais os lugares isolados Rel_Flag, N_A e DF na camada de configuração.

Considerando as distribuições de falha e reparo como exponenciais, os elementos da rede de Petri do bloco básico mostrado na Figura 4.18 são descritos conforme as Tabelas 4.7, 4.8 e 4.9.

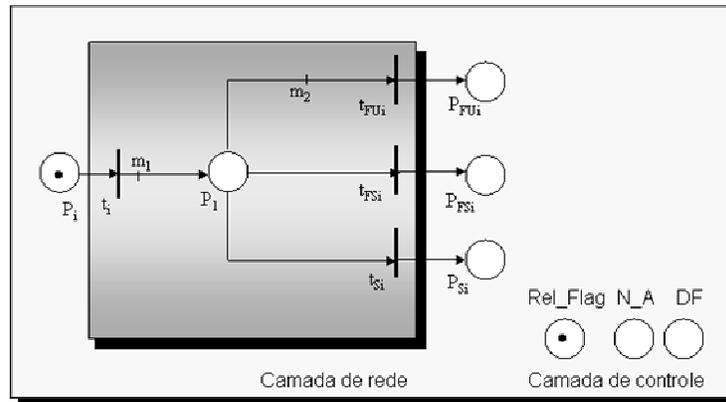


Figura 4.18 Modelo MDP do bloco básico funcional

Tabela 4.7 Lugares do modelo MDP correspondente ao bloco funcional básico

Lugares	Definição
P_i	Lugar de entrada do modelo MDP correspondente à entrada do bloco básico
P_1	Resultado da execução do bloco
P_{Si}	Condição operacional do bloco
P_{FSi}	Bloco não operacional com detecção de falha
P_{Fui}	Bloco não operacional sem detecção de falha
Rel_Flag	Indica o tipo de avaliação a ser executada: #Rel_Flag = 0 → Avaliação de disponibilidade #Rel_Flag = 1 → Avaliação de confiabilidade e segurança
N/A	Indica a forma de obtenção das expressões do bloco #N/A = 0 → Expressões Analíticas
DF	Indica a possibilidade de detecção da falha #DF = 0 → Mascaramento de falha (cobertura perfeita CF=1) #DF = 1 → Detecção de falha dependente do fator de cobertura CF.

As expressões analíticas de confiabilidade e de disponibilidade, mostradas na Tabela 4.8, são obtidas através do modelo EDSPN do nível hierárquico 4, descrito na Figura 4.15.

O modelo da Figura 4.18 também poderá ser aplicado a um bloco de subsistema, quando as funções de confiabilidade e de disponibilidade puderem ser analiticamente definidas. Quando não for possível a obtenção dessas funções analiticamente, então se utiliza o modelo da Figura 4.19 para representação do modelo MDP do bloco de subsistema.

Tabela 4.8 Transições do modelo MDP correspondente ao bloco funcional básico

Trans.	Peso: Expressões lógicas condicionais	Descrição
t_i	1	Execução do bloco básico
t_{Si}	Se #Rel_Flag=1 e #N/A=0: $(\exp^{-\lambda t})$ Se #Rel_Flag=0 e #N/A=0: $\left(\frac{\mu}{\mu + \lambda}\right)$	Expressão analítica de confiabilidade Expressão analítica da disponibilidade
t_{Fi}	Se #Rel_Flag=1 e #N/A=0: $1 - (\exp)^{-\lambda t}$ Se #Rel_Flag=0 e #N/A=0: $\left(\frac{\lambda}{\mu + \lambda}\right)$	Expressão analítica da inconfiabilidade Expressão analítica da indisponibilidade
t_{FSi}	Se #DF=1: CF Se #DF=0: 1	Probabilidade de detecção de falha = CF Probabilidade de detecção = 1
t_{FUi}	(1-CF)	Probabilidade de não detecção da falha

Tabela 4.9 Expressões de multiplicidade de arcos do bloco funcional básico no MDP

Arcos	Expressões lógicas da multiplicidade dos arcos dependentes da marcação	Descrição
m_1	Se #N/A=0: 1	Expressões analíticas correspondentes aos atributos de confiabilidade e disponibilidade do bloco básico
m_2	Se #DF = 1: 1 Se #DF = 0: 2	Detecção de falha (transição habilitada) Mascaramento (transição inabilitada)

O modelo MDP do bloco de subsistema permite a obtenção de valores numéricos de estado permanente para as estimativas de disponibilidade, confiabilidade e segurança, conforme as transições e expressões de multiplicidade de arcos das Tabelas 4.10 e 4.11, respectivamente. Como estimativas de confiabilidade e segurança são necessariamente transientes e o modelo MDP só permite análise de estado permanente [51][52][53], pode-se montar a curva transiente de confiabilidade, do tempo zero até o tempo t , através da análise de estado permanente para um número qualquer de intervalos de tempo, em intervalos regulares, entre 0 e t . A esses intervalos de tempos regulares é associada a variável IT, a qual é um número inteiro divisor de t . Os pontos de verificação, múltiplos de IT, variam de 1, para $t=IT$, até um máximo t/IT .

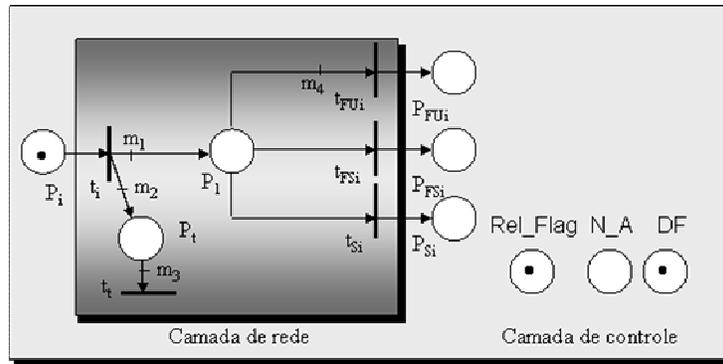


Figura 4.19 Modelo MDP do bloco de subsistema

Por exemplo, considerando-se um intervalo de tempo de 8760 horas, equivalente a 1 ano, pode-se definir 30 pontos de verificação de confiabilidade, falha segura e falha insegura, em intervalos de tempo de $IT=292$ minutos no modelo EDSPN do nível hierárquico 4, e transferir esses valores para o modelo de subsistema no nível hierárquico 5. Uma aproximação mais refinada exige um maior número de pontos de verificação. O disparo da transição t_i indica o ponto de verificação do tempo que está sendo analisado. Ele define a quantidade de *tokens* em P_t , conforme o peso do arco m_2 , na Tabela 4.11, o qual é dado por t/IT . Por exemplo, a Tabela 4.12 considera um intervalo de tempo de 1460 minutos, um intervalo de 292 minutos entre pontos de verificação e valores arbitrários de confiabilidade e disponibilidade obtidos no nível hierárquico 4. O número de *tokens* em P_t determina os valores de confiabilidade correspondentes aos pontos de verificação de um dado bloco de subsistema.

Tabela 4.10 Transições do modelo MDP correspondente ao bloco de subsistema

Trans.	Peso : Expressões lógicas/Valores	Descrição
t_i	1	Execução do bloco de subsistema
t_{Si}	Se $\#Rel_Flag = 1$ e $\#N/A=1$: Valores numéricos de confiabilidade Se $\#Rel_Flag = 0$ e $\#N/A=1$: Valor numérico de disponibilidade de estado permanente	Valores numéricos dos pontos de verificação de confiabilidade entre 0 e t. Valor numérico de disponibilidade.
T_{FSi}	Se $\#Rel_Flag = 1$ e $\#N/A=1$: Valores numéricos de falhas seguras Se $\#Rel_Flag = 0$ e $\#N/A=1$: Valor numérico de falha segura em estado permanente	Valores numéricos dos pontos de verificação de falhas seguras entre 0 e t. Valor numérico de falha segura em estado permanente
t_{Fui}	Se $\#Rel_Flag = 1$ e $\#N/A=1$: Valores numéricos de falhas inseguras Se $\#Rel_Flag = 0$ e $\#N/A=1$: Valor numérico de falha insegura em estado permanente	Valores numéricos dos pontos de verificação de falhas inseguras entre 0 e t. Valor numérico de falha insegura em estado permanente

Tabela 4.11 Expressões de multiplicidade de arcos do modelo MDP do subsistema

Arcos	Expressões lógicas da multiplicidade dos arcos	Descrição
m ₁	1	Habilita a execução do modelo correspondente ao bloco
m ₂	Se #Rel_Flag=1 e #N_A=1: (t / IT) Senão 0	Define o ponto de verificação para análise de confiabilidade numérica por meio da multiplicidade do arco Desabilita análise numérica
m ₃	#P _t	Elimina os <i>tokens</i> correspondentes aos pontos de verificação.
m ₄	Se #DF=1: 1 Senão 2	Deteção de falha (transição habilitada) Mascaramento (transição inabilitada)

Tabela 4.12 Relação número de *tokens* e pontos de verificação

Ponto de Verificação	tempo	Confiabilidade hipotética (em %)	Disponibilidade hipotética (em %)	Número de <i>tokens</i> em P _t
1	292 min	0.90	0.95	1
2	584 min	0.85		2
3	876 min	0.82		3
4	1168 min	0.75		4
5	1460 min	0.63		5

Todos os blocos de um diagrama de sistema devem ter suas estimativas avaliadas para os mesmos pontos de verificação. Logo, se um dos blocos do sistema tem 5 pontos de verificação de tempo, conforme mostrado na Tabela 4.12, os demais blocos do sistema devem calcular expressões numéricas ou analíticas para os mesmos pontos de verificação, de modo que se possa utilizar as estimativas numéricas e analíticas conjuntamente. Com base nos dados arbitrados na Tabela 4.12 e na Figura 4.19, as transições em conflito t_{Si} e t_{Fi} podem ser definidas conforme a Tabela 4.13.

Deste modo, pode-se definir o diagrama do sistema, por meio da conexão dos modelos dependáveis e parametrizados, combinando-se funções analíticas e estimativas numéricas, por meio dos *flags* e parâmetros estruturais.

Para que se possa permitir configurações concorrentes entre blocos dependáveis e parametrizados, são necessárias a existência de blocos de decisão. Esses blocos são representados na Figura 4.20. Os blocos que compõem o nível hierárquico 5 serão descritos mais detalhadamente no próximo capítulo. Eles estão sendo apresentados a priori apenas para possibilitar a exemplificação da metodologia por níveis hierárquicos.

Tabela 4.13 Valores numéricos de confiabilidade para um bloco de subsistema

Trans.	Peso : Expressões lógicas/Valores	Descrição
t_i	1	Execução do bloco de subsistema
t_{Si}	Se Rel_Flag =1 e N/A_Flag=1: Se #P1=1: 0.90 Se #P1=2: 0.85 Se #P1=3: 0.82 Se #P1=4: 0.75 Se #P1=5: 0.63 Se Rel_Flag =0 e N/A_Flag=1: 0.95	Valores numéricos de confiabilidade nos pontos de verificação Valor numérico da disponibilidade
t_{FSi}	Se Rel_Flag =1 e N/A_Flag=1: Se #P1=1: 0.09 Se #P1=2: 0.135 Se #P1=3: 0.162 Se #P1=4: 0.225 Se #P1=5: 0.243 Se Rel_Flag =0 e N/A_Flag=1: 0.05	Valores numéricos de falha segura nos pontos de verificação, para CF=0.90. Valor numérico de indisponibilidade para falhas detectáveis.
t_{FUi}	Se Rel_Flag =1 e N/A_Flag=1: Se #P1=1: 0.01 Se #P1=2: 0.015 Se #P1=3: 0.018 Se #P1=4: 0.025 Se #P1=5: 0.027 Se Rel_Flag =0 e N/A_Flag=1: 0.05	Valores numéricos de falha insegura nos pontos de verificação para (1-CF)=0.10. Valor numérico de indisponibilidade para falhas não detectáveis.

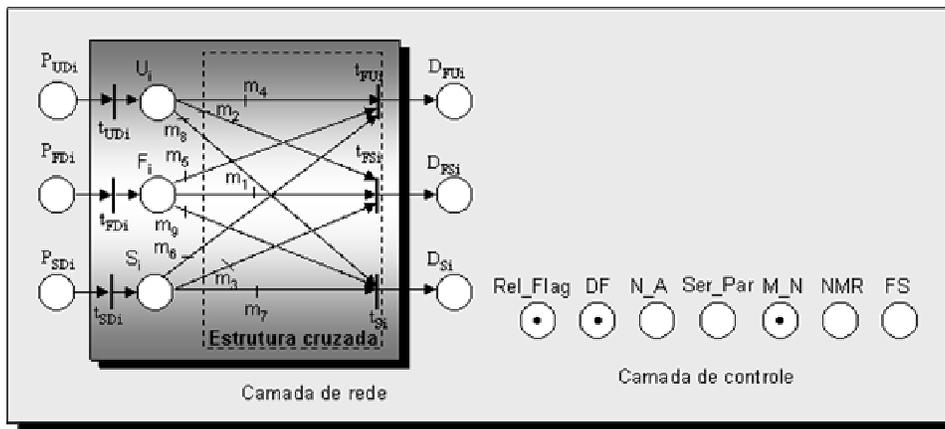


Figura 4.20 Modelo MDP do bloco de decisão

Os blocos de decisão permitem a validação dos resultados de n blocos de entrada, de acordo com regras de decisão. Assim, como os demais blocos dependáveis e parametrizados, este bloco é composto por duas camadas: a) Camada de modelo a qual é responsável pelo refinamento das redes de Petri bem formadas, ou blocos intermediários,

do nível hierárquico 3, por meio de elementos da rede de Petri como lugares, transições, *tokens* e arcos; b) Camada de configuração – composta por *flags* e regras de decisão, que, quando aplicadas aos arcos da estrutura cruzada e aos arcos de entrada do bloco, permitem a liberação de diferentes resultados de saída de acordo com a configuração. A aplicação da metodologia poderá ser melhor compreendida, através do exemplo ilustrativo seguinte.

4.5 Exemplo ilustrativo da aplicação da metodologia

Considere-se a especificação de um sistema cuja estrutura não seja perfeitamente serial/paralela, conforme descrito em [140] e mostrado na Figura 4.21. Em geral, quando a estrutura não se comporta exatamente da forma serial-paralelo, a análise de diagrama de blocos torna-se tediosa, uma vez que os eventos não são mutuamente exclusivos. No exemplo originalmente proposto em [140], as estimativas de confiabilidade são calculadas analiticamente pelo uso do teorema das probabilidades totais. Neste exemplo, pretende-se não apenas avaliar as estimativas de confiabilidade do sistema como um todo, como também as estimativas de disponibilidade e segurança, de um modo prático, com o auxílio da metodologia proposta. A seguir são aplicados os passos da metodologia para obtenção das estimativas de dependabilidade:

Passo 1) Definição do problema, definição dos requisitos de avaliação e especificação do sistema por meio de diagrama de blocos:

1.1) Problema: Pretende-se avaliar a confiabilidade, a disponibilidade e a segurança de um sistema composto por 5 blocos funcionais básicos, isto é, com distribuição de falha e reparo Markovianas, configurados por meio de uma estrutura não-serial/não-paralela, conforme descritos na Figura 4.21. Para possibilitar as avaliações propostas, o sistema deverá funcionar convenientemente entre os pontos x e y , ou seja, deve haver pelo menos um caminho entre x e y que permita ao sistema continuar em operação, ou ainda que o sistema possa falhar e se recuperar com rapidez, e que a maior parte das falhas sejam cobertas.

1.2) Requisitos de avaliação: Confiabilidade, disponibilidade e segurança.

1.3) Especificação do sistema por meio de um diagrama de blocos:

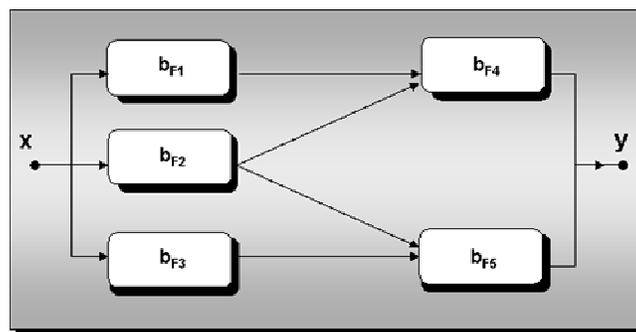


Figura 4.21 Diagrama de blocos

Passo 2) Geração do diagrama de blocos de dependabilidade estendida:

Neste caso, estão sendo considerados apenas os blocos funcionais básicos, de modo a manter fidelidade com o exemplo original. Este diagrama, por meio de sua estrutura redundante e paralela, é naturalmente um diagrama tolerante a falhas. Conforme observado na Figura 4.21 os blocos b_{F4} e b_{F5} possuem dois caminhos de entrada por meio dos blocos b_{F1} e b_{F2} , e b_{F2} e b_{F3} , respectivamente. Neste caso, nas entradas dos blocos b_{F4} e b_{F5} são colocados blocos de decisão, b_{d1} e b_{d2} , os quais estão em série com esses blocos. Para o nó y , o qual é a saída do sistema, também convergem dois caminhos distintos através dos blocos b_{F4} e b_{F5} . Neste caso, substitui-se o nó y , por um bloco de decisão, b_{d3} , em série com o nó y . O diagrama de blocos de dependabilidade estendida, EDBD correspondente ao diagrama de blocos da Figura 4.21, é mostrado na Figura 4.22.

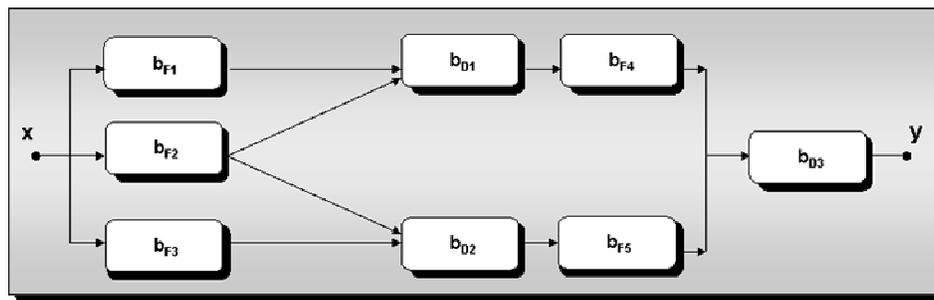


Figura 4.22 Diagrama EDBD do diagrama de blocos do sistema

O diagrama EDBD, por meio dos blocos de decisão, permite a estruturação de sistemas serial/paralelo ou não-serial/não-paralelo, para que as avaliações possam ser feitas de um modo mais flexível no nível hierárquico 5.

Passo 3) Geração dos modelos intermediários (por meio das regras de composição modular):

Através da aplicação dos métodos de composição, definidos em 4.4.3.1, aos blocos intermediários (redes de Petri de alto nível), por meio de junção, fusão ou adição, obtém-se a representação do diagrama de sistema intermediário, como mostra a Figura 4.23. Para que esse diagrama possa ser analisado quantitativamente, deve-se proceder ao refinamento de cada bloco individualmente de modo que possam ser definidas estimativas analíticas ou numéricas.

Passo 4) Geração dos modelos EDSPN e avaliações analíticas ou numéricas dos atributos de dependabilidade.

Sendo todos os blocos, que compõem o diagrama EDBD da Figura 4.22, básicos, as expressões analíticas de cada bloco se comporta de acordo com a Tabela 4.1, correspondente à Figura 4.15. Uma vez que as expressões analíticas do modelo EDSPN são conhecidas, estas são transferidas para os modelos MDP no nível hierárquico 5.

Passo 5) Geração dos modelos MDP para os blocos individuais e geração do diagrama de sistema dependável e parametrizado.

No passo 5, refina-se cada bloco intermediário mostrado na Figura 4.23, de acordo com as informações obtidas dos modelos EDSPN. Os modelos MDP, dos blocos básicos neste nível de representação, mostrados na Figura 4.18, são determinados por meio das Tabelas 4.7, 4.8 e 4.9, enquanto o bloco de decisão é mostrado na Figura 4.20.

As transições temporizadas $T_{Feedback1}$, $T_{Feedback2}$ e $T_{Feedback3}$, denominadas transições de *feedback*, são utilizadas com o propósito de permitir a análise de estado permanente do modelo [51][52][53]. Os valores dos tempos de retardo das transições de *feedback* são iguais, entre si, e maiores do que zero. As demais transições do modelo, sendo imediatas, isto é, tempo de retardo igual a zero, faz com que o *token*, ao chegar aos lugares P_{Si} , P_{FSi} e P_{FUi} tenha um tempo de permanência infinito se comparado com o retardo zero das transições imediatas, simulando dessa forma um pseudo estado absorvente, numa análise de probabilidade por meio de tempos relativos.

As métricas de dependabilidade do sistema em geral, são definidas pelas expressões mostradas na Tabela 4.14.

Tabela 4.14 Métricas do sistema dependável e parametrizado

Atributos	Métrica
Confiabilidade	$P\{\#D_{Sp}=1\}$
Inconfiabilidade	$P\{\#D_{FUp}+\#D_{FSp}=1\}$
Falha Segura	$P\{\#D_{FSp}=1\}$
Falha Insegura	$P\{\#D_{FUp}=1\}$
Segurança	$P\{\#D_{Sp}+\#D_{FSp}=1\}$
Insegurança	$P\{\#D_{FUp}=1\}$
Disponibilidade	$P\{\#D_{Sp}=1\}$
Indisponibilidade	$P\{\#D_{Sp}=0\}$

O modelo MDP, do diagrama de sistema correspondente ao diagrama EDBD da Figura 4.22, é mostrado na Figura 4.24. Nesse diagrama, pode-se observar a troca de cada bloco intermediário, da Figura 4.23, pelo modelo MDP correspondente. Na Figura 4.24, apenas os *flags* das camadas de configuração relevantes ao problema são especificados para cada bloco. Como a análise é executada sobre o modelo MDP do diagrama de sistema, e como o parâmetro *Rel_Flag* deve ser o mesmo para todos os blocos que compõem este modelo, somente é necessária a sua definição uma única vez no modelo. Ou seja, caso o parâmetro *Rel_Flag* esteja ativado, o modelo estima a confiabilidade do sistema, caso o parâmetro *Rel_Flag* esteja desativado, o modelo estima a disponibilidade.

Considerações Finais

Neste capítulo, a metodologia de modelagem e análise de sistemas dependáveis é estabelecida partindo-se dos conceitos de modularização e hierarquia. De modo a simplificar a análise, os sistemas são divididos em blocos, os quais são avaliados em diferentes níveis hierárquicos. Os níveis hierárquicos diferem entre si em função do grau de detalhes associados a cada bloco. A metodologia de modelagem pode ser usada para sistemas em configurações serial, paralela, m/n, NMR e *flux-summing*, além de configurações não-serial/não-paralela [126] ou serial/paralela complexa [140]. Os modelos dependáveis e parametrizados são representados por meio de duas camadas: a) camadas de modelo; b) camadas de configuração. A camada de configuração, representada por um conjunto de lugares isolados, permite a reusabilidade dos modelos dos blocos que compõem um sistema, em outros sistemas, além de flexibilizar a adoção de diferentes configurações desses blocos no sistema. A utilização de expressões analíticas e numéricas nos modelos MDP, além da definição de regras de composição, possibilita a análise de dependabilidade de sistemas complexos. Nesta metodologia, parâmetros de configuração e estruturais, bem como algumas características dos modelos EDSPN e MDP dependentes de marcação, tais como, multiplicidade condicional dos arcos, multiplicidade condicional dos tempos médios de retardo das transições temporizadas e multiplicidade de pesos associados às transições imediatas, por meio de expressões analíticas ou numéricas, são utilizados de forma intensiva. Esta metodologia lida de um modo satisfatório com a geração do espaço de estados e a característica de *stiffness*, através da geração de um espaço de estados limitado e solução baseada na forma produto, a qual é expressa como um produto de fatores descrevendo o estado de cada bloco.

Capítulo 5

Especificação dos Modelos

Introdução

Os modelos apresentados neste capítulo, de acordo com a estrutura hierárquica anteriormente definida, estão relacionados aos níveis hierárquicos 4 e 5. A construção de uma biblioteca de modelos básicos permite ao modelador a simplificação do processo de modelagem dos sistemas dependáveis. Os modelos EDSPN do nível hierárquico 4, têm por objetivo um maior detalhamento de cada bloco individual do diagrama intermediário do sistema, com respeito aos atributos de confiabilidade, disponibilidade e segurança, e a geração de expressões analíticas ou numéricas utilizadas no nível hierárquico 5, através dos modelos MDP. As expressões condicionais de multiplicidade de arcos e dos tempos de retardo das transições temporizadas permitem diferentes possibilidades de modelagem aos modelos EDSPN tornando a análise dos diagramas de sistemas mais flexível, além de possibilitar o reuso dos modelos obtidos em outros diagramas.

Não apenas as estimativas de dependabilidade geradas analiticamente, ou obtidas numericamente por meio de avaliações sucessivas no modelo EDSPN, como também as expressões analíticas de dependabilidade, definidas formalmente na literatura por meio de funções de distribuição de probabilidade, podem ser utilizadas diretamente nos modelos MDP. Caso as expressões analíticas não sejam conhecidas ou sejam de difícil tratamento analítico, pode-se obter as estimativas do modelo por meio de avaliações numéricas de estado permanente, para um determinado número de pontos de verificação em intervalos regulares de tempo. A quantidade de pontos de verificação a serem utilizados é função da aproximação desejável das funções de distribuição associadas aos atributos de dependabilidade requeridos.

Os modelos MDP, definidos mais adiante, objetivam contextualizar os modelos EDSPN correspondentes a blocos individuais do nível hierárquico 4 em um ambiente de sistema dependável, através da utilização das estimativas de dependabilidade analíticas ou numéricas obtidas, e pelo acréscimo de parâmetros estruturais e de controle. Os modelos MDP estruturados na forma produto, permitem a obtenção das estimativas de dependabilidade do sistema como um todo.

5.1 Nível Hierárquico 4 - Modelos EDSPN

Os modelos EDSPN apresentados correspondem as diversas formas de redundância descritas no nível hierárquico 1 do Capítulo 4 [109][3]. Os modelos podem ser dos seguintes tipos:

- bloco básico sem replicação;
- bloco básico com replicação passiva do tipo *coldstandby*;
- bloco básico com replicação passiva do tipo *warmstandby*;
- bloco básico com replicação semi-ativa do tipo *hotstandby*;
- bloco básico com replicação ativa NMR;
- bloco básico com replicação ativa generalizada.

Estes modelos fazem parte da biblioteca de modelos básicos, correspondentes as diversas formas de replicação do bloco básico, a partir da qual o desenvolvimento de novos modelos torna-se possível.

5.1.1 Modelo EDSPN – Bloco Básico sem Replicação

O modelo EDSPN correspondente ao bloco básico sem replicação não apresenta qualquer forma de redundância, sendo representado apenas pelo bloco ativo cujas distribuições de falha e reparo são exponencialmente distribuídas. Este modelo, um grafo com anotações de lugares e transições, tem o tempo médio para falha, MTTF, associado à transição Falha_BB, e o tempo médio de reparo, MTTR, associado à transição Reparo_BB. Sendo as distribuições de falha e reparo exponenciais, pode-se representar o retardo das mesmas pelos parâmetros MTTF e MTTR, ou pelo inverso das taxas de falha e reparo, λ e μ , respectivamente, conforme descrito na Figura 5.1. Um *token* no lugar BB_ON, indica que o bloco encontra-se operacional ou ativo, enquanto um *token* no lugar BB_OFF, indica que o bloco encontra-se inativo ou inoperante. Caso haja mais de um bloco básico num diagrama EDBD, as denominações dos lugares, transições, tempos médios e taxas de falha e reparo, em cada modelo, deverão ser exclusivas.

Na Figura 5.1 o arco de entrada direcionado do lugar BB_OFF à transição Reparo_BB, apresenta uma multiplicidade condicional de marcação, e está condicionada à presença ou ausência de *token* no lugar de controle Rel_Flag. A presença de *token* no lugar Rel_Flag permite o cálculo da confiabilidade do bloco básico, enquanto a ausência de *token* nesse lugar permite a computação da disponibilidade. A multiplicidade condicional de arco dependente da marcação está representada por um rótulo e um pequeno traço transversal sobre o arco. A transição temporizada Falha_BB será disparada, quando houver um token em BB_ON, significando que o bloco correspondente apresenta-se em falha. A transição Reparo_BB será disparada quando o bloco em falha tiver sido recuperado.

As avaliações de dependabilidade apresentadas nos estudos de casos, consideram intervalos de tempo de 1 ano, ou 8760 horas, para aplicações de telecomunicações, ou de apenas 3 horas para aplicações na área de aviação. A variabilidade do intervalo de tempo é dependente do tipo de aplicação.

As expressões analíticas para confiabilidade e disponibilidade a serem utilizadas nos modelos MDP, bem como as métricas para obtenção de estimativas numéricas, são apresentadas na Tabela 5.1.

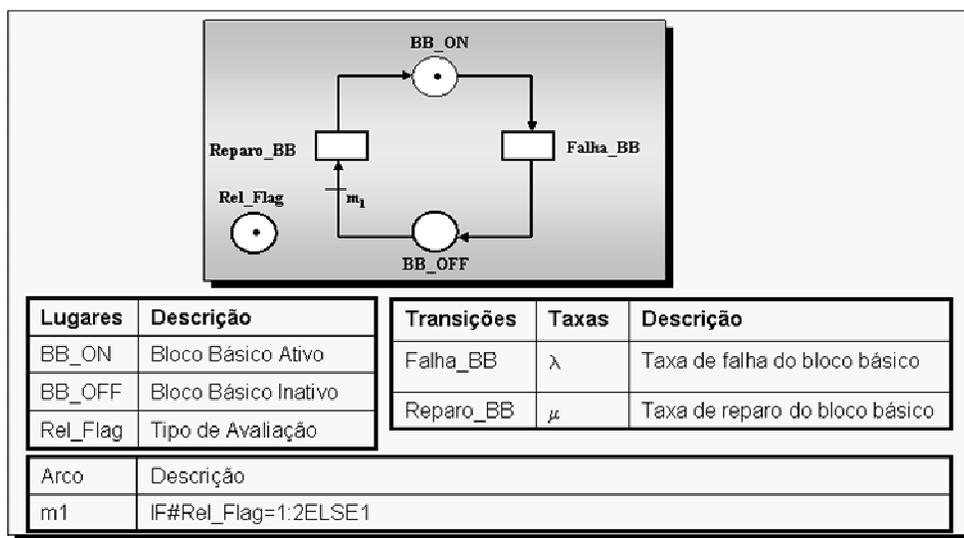


Figura 5.1 Modelo EDSPN para bloco básico sem replicação

Tabela 5.1 Expressões analíticas e métricas de dependabilidade

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \exp^{-\lambda t}$	$P\{\#BB_ON>0\}$
Inconfiabilidade	$Q(t) = (1 - \exp^{-\lambda t})$	$P\{\#BB_ON=0\}$
Disponibilidade	$A = \frac{1}{1 + \frac{\lambda}{\mu}} = \frac{\mu}{\mu + \lambda}$	$P\{\#BB_ON>0\}$
Indisponibilidade	$U = 1 - \frac{1}{1 + \frac{\lambda}{\mu}} = \frac{\lambda}{\mu + \lambda}$	$P\{\#BB_ON=0\}$

Onde a expressão $P\{\#BB_ON>0\}$ indica a probabilidade do bloco básico sem replicação estar numa condição confiável ou operacional. A condição $P\{\#BB_ON=0\}$ indica a probabilidade do bloco básico sem replicação estar numa condição de falha ou não operacional. Esta mesma condição também pode ser expressa pela indicação da presença de *token* no lugar BB_OFF.

5.1.2 Modelo EDSPN – Bloco Básico com Replicação Passiva *ColdStandby*

Este modelo de replicação considera o bloco *standby* como sendo formado por $n \geq 2$ blocos, sendo que apenas um deles está ativo, ou operacional, enquanto os demais estão inativos, ou inoperantes. O bloco ativo processa os dados de entrada e está sujeito a falhas. Considera-se que as demais réplicas passivas encontram-se inativas e desenergizadas. As réplicas passivas não estão sujeitas a falhas e nem processarão os dados de entrada, enquanto permanecerem na condição passiva. As réplicas passivas permanecem na condição de espera até a ocorrência de falha no componente ativo, quando então um dos componentes passivos é comutado para a condição ativa e vice-

versa. O modelo EDSPN correspondente ao bloco básico com replicação passiva do tipo *coldstandby* (BCS), é apresentado na Figura 5.2.

Na Figura 5.2 os arcos de entrada e de saída da transição *Reparo_BCS* apresentam multiplicidades condicionais de marcação representadas pelos rótulos m_1 , m_2 e m_3 , e um pequeno traço transversal sobre o respectivo arco. As correspondentes expressões lógicas condicionais são apresentadas também na Figura 5.2. Considerando-se que todos os blocos que compõem o modelo *coldstandby* tenham distribuições exponenciais de falha e reparo, com taxas constantes λ e μ , respectivamente, podem-se definir as expressões analíticas para confiabilidade e disponibilidade a serem utilizadas nos modelos MDP de acordo com a Tabela 5.2. Observa-se que a expressão de confiabilidade do modelo *coldstandby* é representada por uma distribuição de Erlang. Na Tabela 5.2 também são definidas as métricas para obtenção de estimativas numéricas dos atributos de dependabilidade.

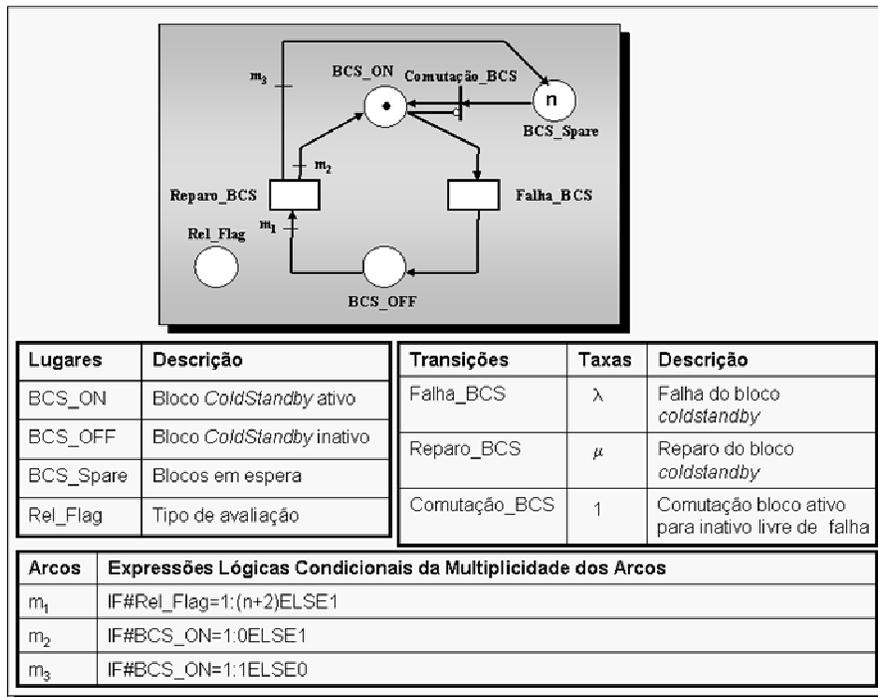


Figura 5.2 Modelo EDSPN para o bloco com replicação passiva *coldstandby*

Como pode ser verificado na Tabela 5.2, as expressões de confiabilidade e disponibilidade consideram as mesmas taxas de falha e de reparo para todas as réplicas, e uma semântica de temporização *single server*, a qual corresponde a possibilidade da ocorrência de uma falha, ou de um reparo, por vez. Em sistemas reais a possibilidade de um único reparo por vez equivale a existência de um único reparador, ou de uma única equipe de manutenção. Nas expressões de confiabilidade e disponibilidade, a variável i indica o número de réplicas passivas do modelo e a variável n o número total de réplicas.

Tabela 5.2 Expressões analíticas e métricas de dependabilidade para o bloco básico com replicação passiva *coldstandby*.

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \sum_{i=0}^{n-1} \frac{(\lambda t)^i}{i!} \exp^{-\lambda t}$	$P\{\#BCS_ON>0\}$
Inconfiabilidade	$Q(t) = 1 - \sum_{i=0}^{n-1} \frac{(\lambda t)^i}{i!} \exp^{-\lambda t}$	$P\{\#BCS_ON=0\}$
Disponibilidade	$A = \left[1 + \sum_{k \geq 1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right) \right]^{-1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right), k \geq 1$	$P\{\#BCS_ON>0\}$
Indisponibilidade	$U = 1 - \left[1 + \sum_{k \geq 1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right) \right]^{-1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right), k \geq 1$	$P\{\#BCS_ON=0\}$

5.1.3 Modelo EDSPN – Bloco Básico Com Replicação Passiva *WarmStandby*

Este bloco é representado pelo mesmo modelo EDSPN do bloco *coldstandby*, anteriormente mostrado na Figura 5.2. O bloco *warmstandby* é composto por $n \geq 2$ réplicas, sendo que apenas uma delas está operacional enquanto as demais estão inativas. O bloco ativo processa os dados de entrada e está sujeito a falhas, enquanto as demais réplicas não processarão os dados de entrada, porém estarão sujeitas a falhas nesse período. As taxas de falha das réplicas passivas são, em geral, menores do que a taxa de falha da réplica ativa, em virtude do menor estresse a que são submetidas. Caso a réplica ativa torne-se inabilitada, esta será substituída por uma das réplicas passivas em espera. A réplica em espera ao assumir a condição ativa, assume também a taxa de falha da réplica ativa. Neste modelo as taxas de falha das réplicas quando em atividade e quando em espera são diferentes. Considerando-se que todos os blocos que compõem o modelo *warmstandby* tenham uma taxa constante de falha λ , quando em atividade, e uma taxa constante de falha ξ quando em espera, e uma taxa constante de reparo μ , as expressões analíticas para confiabilidade e disponibilidade utilizadas nos modelos MDP têm suas definições de acordo com a Tabela 5.3, a qual também apresenta as expressões correspondentes às métricas de dependabilidade requeridas.

Na utilização do modelo da Figura 5.2, considera-se as taxas $v_1, v_2, \dots, v_{(n-1)}$ e $v_{(n)}$ ao invés das taxas $\lambda_1, \lambda_2, \dots, \lambda_{(n-1)}, \lambda_n$ na expressão de confiabilidade, onde $v_1 = \lambda + (n-1)\xi$, $v_2 = \lambda + (n-2)\xi, \dots, v_{(n-1)} = \lambda + \xi, v_n = \lambda$, correspondem ao somatório das taxas de falha da réplica ativa e das réplicas passivas num determinado instante de tempo, uma vez que a distribuição do modelo corresponde a uma distribuição hipoexponencial.

A expressão de confiabilidade representada pela distribuição hipoexponencial, é mostrado na Figura 5.3. Esta distribuição corresponde a uma distribuição de estágios de Cox [140] para um sistema redundante com $(n-1)$ blocos em espera e um bloco ativo. Considera-se, neste caso, que a cobertura de falhas seja perfeita, ou seja, que todas as

ocorrências de falha sejam detectadas. Caso as coberturas de falhas, em cada estágio da distribuição de Cox, não sejam perfeitas, porém semelhantes e iguais a C para todas as réplicas, a representação do modelo por meio da distribuição de estágios de Cox, será tratada conforme a Figura 5.4. Observa-se que este tipo de representação é conveniente para análise e entendimento do bloco com replicação passiva *warmstandby*.

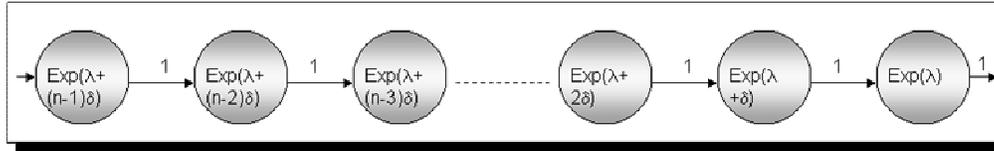


Figura 5.3 Distribuição do tempo de vida do bloco básico *warmstandby* com cobertura perfeita no modelo de estágios de Cox

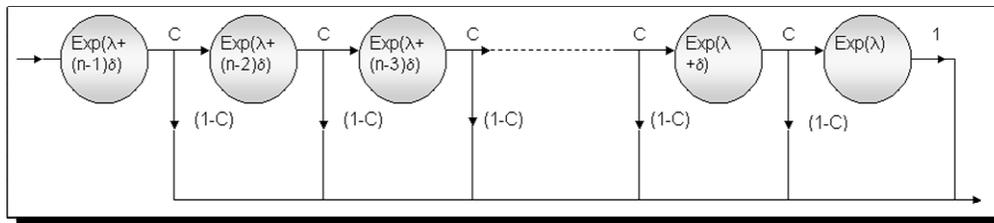


Figura 5.4 Distribuição do tempo de vida do bloco básico *warmstandby* com cobertura imperfeita

Tabela 5.3 Expressões analíticas e métricas de dependabilidade para o bloco básico com replicação passiva *warmstandby*.

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \sum_{i=1}^n \left[\prod_{\substack{j=1 \\ j \neq i}}^n \frac{v_j}{v_j - v_i} \right] (\exp^{-v_i t})$	$P\{\#BCS_ON > 0\}$
Inconfiabilidade	$Q(t) = 1 - \left[\sum_{i=1}^n \left[\prod_{\substack{j=1 \\ j \neq i}}^n \frac{v_j}{v_j - v_i} \right] (\exp^{-v_i t}) \right]$	$P\{\#BCS_ON = 0\}$
Disponibilidade	$A = \left[1 + \sum_{i=0}^{n-1} \prod_{j=0}^i \frac{\lambda + (n-j-1)\delta}{\mu(j+1)} \right]^{-1} \left(\prod_{i=0}^{n-1} \frac{\lambda + (n-i-1)\delta}{\mu(i+1)} \right)$ $\left(\prod_{j=m}^{n-1} \frac{\lambda + (n-j-1)}{m\mu} \right)$	$P\{\#BCS_ON > 0\}$
Indisponibilidade	$U = 1 - \left[1 + \sum_{i=0}^{n-1} \prod_{j=0}^i \frac{\lambda + (n-j-1)\delta}{\mu(j+1)} \right]^{-1} \left(\prod_{i=0}^{n-1} \frac{\lambda + (n-i-1)\delta}{\mu(i+1)} \right)$ $\left(\prod_{j=m}^{n-1} \frac{\lambda + (n-j-1)}{m\mu} \right)$	$P\{\#BCS_ON = 0\}$

Nas expressões de disponibilidade considera-se m como sendo o número de reparadores. Caso haja apenas um único reparador, usa-se a semântica de temporização *single server*; caso m seja tal que para cada bloco em reparo tenha um reparador exclusivo, usa-se a semântica de temporização *infinite server*. No modelo EDSPN pode-se definir um número qualquer de reparadores pela utilização da semântica *single server*, desde que a expressão de *delay* do reparo seja uma função condicional dependente do parâmetro estrutural m , correspondente ao número de reparadores, como pode ser visto nas expressões de disponibilidade e indisponibilidade mostradas na Tabela 5.3. Neste caso, para um conjunto de n blocos, podem-se utilizar um número m de reparadores, numa semântica de temporização *multiple server*.

5.1.4 Modelo EDSPN – Bloco Básico Com Replicação Semi-Ativa *HotStandby*

No modelo de bloco básico com replicação semi-ativa do tipo *hotstandby*, as réplicas estão ativas e processando simultaneamente as entradas. O resultado de saída, contudo, é liberado por apenas uma das réplicas, denominada réplica primária. As demais réplicas, denominadas secundárias, não liberam qualquer resultado para a saída. Caso a réplica ativa primária venha a falhar, a réplica primária é trocada por uma das réplicas secundárias, a qual passará a exercer a função da réplica primária. Neste caso, como as réplicas secundárias mantêm a evolução do contexto corrente, a nova réplica é rapidamente e automaticamente inicializada. Os eventos de falha e reparo das réplicas são independentes e exponencialmente distribuídos, com parâmetros λ e μ , respectivamente. A Figura 5.5 representa o modelo do bloco básico do tipo *hotstandby*, o qual é bastante similar ao modelo do bloco básico sem replicação mostrado na Figura 5.1, exceto pelo número n de réplicas e pela semântica *infinite server* de temporização das transições temporizadas ao invés da semântica *single server*. A semântica de temporização *infinite server*, permite que todas as réplicas sejam ativadas simultaneamente e seus tempos de vida sejam decrementados até zero, em paralelo. A falha do bloco ocorre se todas as réplicas falharem. As expressões de confiabilidade e de disponibilidade obtidas para o modelo do bloco *hotstandby* é similar as expressões obtidas para blocos em paralelo.

Tabela 5.4 Expressões analíticas e métricas de dependabilidade para modelo *hotstandby*

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = 1 - (1 - \exp^{-\lambda t})^n$	$P(t > T) = P\{\#BHS_ON > 0\}$
Inconfiabilidade	$Q(t) = (1 - \exp^{-\lambda t})^n$	$P(t \leq T) = P\{\#BHS_ON = 0\}$
Disponibilidade	$A = \left(1 - \left(1 - \frac{1}{1 + \frac{\lambda}{\mu}} \right)^n \right)$	$P\{\#BHS_ON > 0\}$
Indisponibilidade	$U = \left(1 - \frac{1}{1 + \frac{\lambda}{\mu}} \right)^n$	$P\{\#BHS_ON = 0\}$

Conforme pode ser visualizado nas expressões de confiabilidade e disponibilidade da Tabela 5.4, considera-se que todas as réplicas tenham as mesmas taxas de falha e de reparo, e que sejam independentes.

As expressões de confiabilidade e disponibilidade, para diferentes taxas de falha e de reparo, não são representadas no modelo EDSPN, podendo no entanto, serem representadas estruturalmente por expressões lógicas condicionais no modelo MDP por meio de blocos básicos sem replicação.

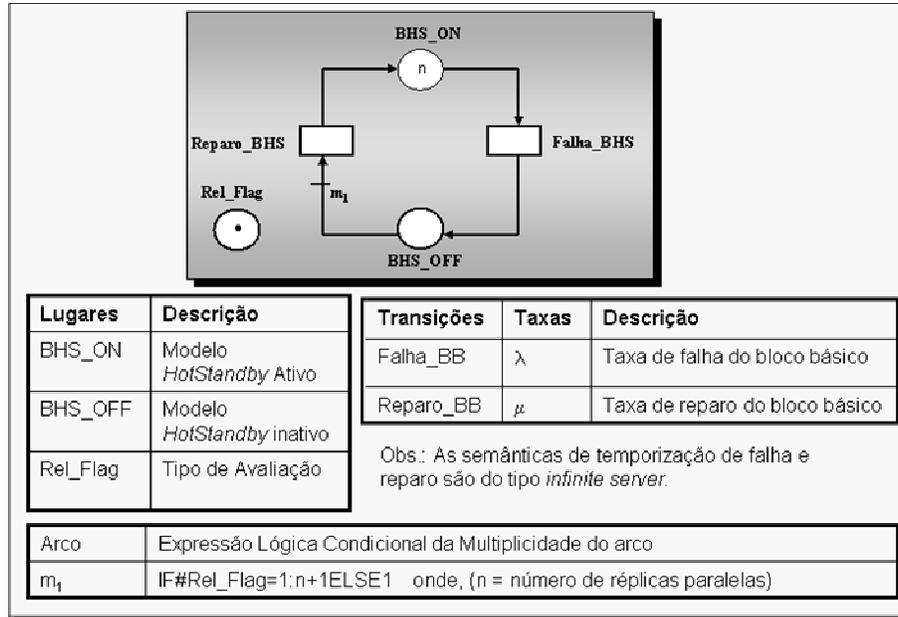


Figura 5.5 Modelo EDSPN para o bloco com replicação semi-ativa *hotstandby*

5.1.5 Modelo EDSPN – Bloco Básico Com Replicação Ativa NMR

No modelo de bloco básico com replicação ativa do tipo *N-Modular Redundancy*, todas as réplicas estão ativas e energizadas, processando as entradas concorrentemente. Considera-se que as réplicas sejam estatisticamente independentes com relação a falhas e reparos. Os resultados são liberados pelas réplicas de um modo síncrono para um mecanismo de decisão, denominado votador. De um total de n réplicas, k réplicas devem estar ativas ou funcionando convenientemente para que o bloco como um todo possa estar operacional. Considerando-se o número de réplicas um número inteiro e ímpar, as ocorrências de falhas serão mascaradas quando houver falhas em $(n-1)/2$ réplicas. Um resultado válido é liberado quando pelo menos $(n+1)/2$ réplicas ativas liberarem resultados válidos. Em aplicações onde haja necessidade de se tolerar pelo menos 1 falha, sem que o bloco NMR como um todo falhe, é interessante a utilização desse tipo de replicação, uma vez que o mascaramento da falha permite uma tolerância a falhas em 100% do tempo. Nas definições dos blocos anteriores, a redundância sendo dinâmica requer que as falhas sejam toleradas por meio de métodos dinâmicos de detecção, localização e reconfiguração os quais para propiciar um nível de tolerância a falha em

100% do tempo, devem requerer um fator de cobertura de falhas de 100%, o que é difícil, senão impossível, de obter [70].

Na Figura 5.6 é apresentado o modelo do bloco básico com replicação ativa do tipo NMR, o qual apresenta similaridades com os modelos de blocos básicos e os modelos de replicação semi-ativa *hotstandby*. No modelo NMR, assim como no modelo *hotstandby*, n é o número de réplicas, e os disparos das transições temporizadas, *Falha_NMR* e *Reparo_NMR* obedecem a uma semântica *infinite server* de temporização ao invés da semântica *single server*, uma vez que as réplicas atuam concorrentemente.

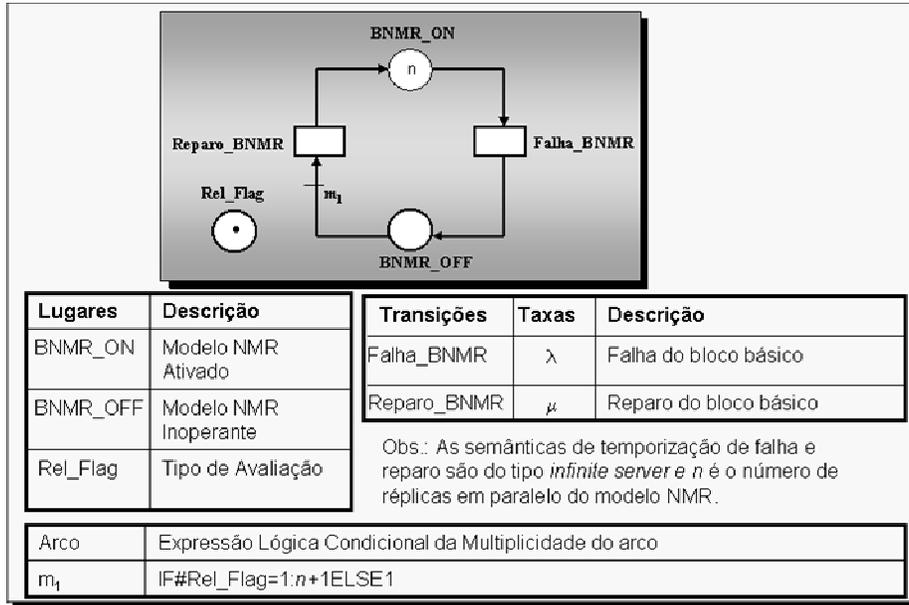


Figura 5.6 Modelo EDSPN para o bloco com replicação ativa *NMR*

Como pode ser observado na Tabela 5.5, as expressões de confiabilidade e de disponibilidade levam em consideração a independência estatística entre as réplicas, mantendo, porém as taxas de falha (λ) e de reparo (μ) semelhantes para todas as réplicas. As expressões de confiabilidade e de disponibilidade para diferentes taxas de falha e de reparo podem ser representadas estruturalmente por expressões lógicas condicionais no modelo MDP utilizando blocos básicos sem replicação, cada qual com taxa de falha (λ_i) e taxa de reparo (μ_i). O modelo EDSPN é utilizado quando as taxas de falha e reparo são similares para todas as réplicas, enquanto o modelo MDP poderá ser utilizado quando as taxas forem similares ou diferentes.

Os modelos EDSPN dos diversos blocos levam em conta as diversas formas de replicação definidas no Capítulo 4. Os modelos EDSPN aplicados aos blocos de forma isolada permitem, de um modo tratável, a obtenção de expressões analíticas ou numéricas desses blocos. Os modelos MDP permitem a incorporação das expressões obtidas através dos modelos EDSPN para possibilitar as estimativas de dependabilidade do sistema como um todo. Por meio dos modelos MDP procura-se contextualizar os diversos blocos individuais num ambiente de sistema.

Tabela 5.5 Expressões analíticas e métricas de dependabilidade para o modelo NMR

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \sum_{k=\binom{n+1}{2}}^n \binom{n}{k} R(t)^k (1-R(t))^{n-k}$	$P\{\#BNMR > (n-1)/2\}$
Inconfiabilidade	$Q(t) = 1 - \sum_{k=\binom{n+1}{2}}^n \binom{n}{k} R(t)^k (1-R(t))^{n-k}$	$P\{\#BNMR_ON < (n+1)/2\}$
Disponibilidade	$A = \sum_{k=\binom{n+1}{2}}^n \binom{n}{k} \left(\frac{1}{1 + \frac{\lambda}{\mu}} \right)^k \left(\frac{\frac{\lambda}{\mu}}{1 + \frac{\lambda}{\mu}} \right)^{n-k}$	$P\{\#BNMR > (n-1)/2\}$
Indisponibilidade	$U = 1 - \sum_{k=\binom{n+1}{2}}^n \binom{n}{k} \left(\frac{1}{1 + \frac{\lambda}{\mu}} \right)^k \left(\frac{\frac{\lambda}{\mu}}{1 + \frac{\lambda}{\mu}} \right)^{n-k}$	$P\{\#BNMR_ON < (n+1)/2\}$

5.2 Nível Hierárquico 5: Modelos Dependáveis e Parametrizados (MDP)

Os modelos MDP, definidos no nível hierárquico 5, podem representar outros tipos de distribuições, não sendo restrito as distribuições exponenciais associadas às transições temporizadas, como nos modelos EDSPN. Necessita-se, entretanto, que as expressões numéricas ou analíticas sejam disponibilizadas. Basicamente os modelos MDP são modelos de redes de Petri, com características específicas, constituídos por:

- **lugares** que descrevem as condições ou situações locais dos estados dos blocos no contexto do sistema;
- **transições** que descrevem os eventos que podem modificar o estado dos blocos de um sistema. Estes eventos podem ser representados por expressões lógicas condicionais dependentes da marcação, os quais são compostos por vários construtores IFs e um ELSE ao final da proposição, conforme sintaxe descrita no ANEXO-B. Transições imediatas em conflito podem conter expressões analíticas ou numéricas correspondentes às funções de distribuição de probabilidade associadas à confiabilidade ou à disponibilidade;
- **arcos** que especificam as relações entre os estados locais dos blocos e os eventos gerados, indicando o estado local em que o evento possa ocorrer, e as transformações de estado local induzidas pelos eventos. As multiplicidades dos arcos no modelo MDP são em geral dependentes das marcações e dos parâmetros

de controle e estrutural, e são definidas por expressões lógicas condicionais, compostas por vários IFs e um ELSE ao final da proposição.

- *tokens* que especificam o estado da rede de Petri. Se um lugar descreve uma condição, a presença de um *token* neste lugar indica que a condição é verdadeira, enquanto a sua ausência indica que a condição é falsa. Se um lugar define uma situação, o número de *tokens* presentes no lugar pode especificar, por exemplo, a quantidade de blocos operando em paralelo ou sendo reparados.

O processo dinâmico de uma rede de Petri é governado por regras de disparo associado às transições, regras essas que neste modelo serão, em muitos casos, definidas por expressões lógicas condicionais dependentes da situação dos estados locais dos blocos. O modelo MDP contém basicamente transições imediatas. Transições temporizadas são colocadas para tornar o modelo de rede de Petri de *loop* fechado, de modo a possibilitar avaliações de estado permanente na forma produto [22]. Na elaboração do diagrama de sistema dependável, os modelos MDP dos blocos individuais podem ser organizados de diversas maneiras: serial, paralela, serial/paralela, com ou sem redundância, com réplicas ativas e passivas, energizadas ou não. Também modelos não-serial/não-paralelo são analisados pelos modelos MDP, bem como formas específicas de paralelismo: m/n, NMR e *flux-summing*, estas duas últimas a serem usadas em estudos de caso. Com o modelo MDP pode-se avaliar analiticamente e/ou numericamente o modelo correspondente ao diagrama de bloco, sem a geração de espaço de estados de grandes dimensões, e sem a característica de *stiffness*.

Os modelos MDP são estruturalmente compostos por duas camadas:

- a) camada de modelo – composta pelos elementos estáticos da rede de Petri, tais como lugares, transições e arcos e pelos elementos dinâmicos da rede representados pelos *tokens*;
- b) camada de configuração – composta por parâmetros que proporcionam o controle do fluxo de *tokens*, através de expressões lógicas nos arcos e transições conflitantes, e por parâmetros que definem a configuração seqüencial ou concorrente dos blocos no modelo.

O modelo MDP não está limitado a distribuições Markovianas. Caso expressões analíticas, ou ainda dados estatísticos históricos ou do fabricante, associados aos atributos de confiabilidade, disponibilidade e segurança sejam acessíveis, os modelos MDP poderão ser usados sem que haja modificação do modelo. Esta flexibilidade, assim como a possibilidade de reuso, são características válidas desse modelo.

A seguir, serão descritos os modelos MDP correspondentes aos blocos constituintes do modelo EDBD, definidos no Capítulo 4. O modelo MDP permite a interligação dos modelos de blocos com expressões analíticas, expressões numéricas ou uma mistura de ambos.

5.2.1 Modelo MDP: Modelo Bloco Básico Markoviano

Este modelo é caracterizado por apresentar distribuições de probabilidade Markoviana para eventos de falha e de reparo. Este modelo, assim como os demais modelos de blocos definidos, apresenta parâmetros estruturais e de configuração, denominados *flags*, cuja função é permitir a parametrização do modelo nas expressões condicionais de multiplicidade dos arcos, dos tempos médios de disparo das transições temporizadas, e dos pesos das transições imediatas em conflito. A parametrização dos modelos além de permitir a sua reutilização, proporciona uma maior flexibilidade de análise dos sistemas. Os modelos dependáveis e parametrizados, estruturados na forma produto, permitem a obtenção das estimativas de dependabilidade do sistema com respeito aos atributos de confiabilidade, disponibilidade e segurança.

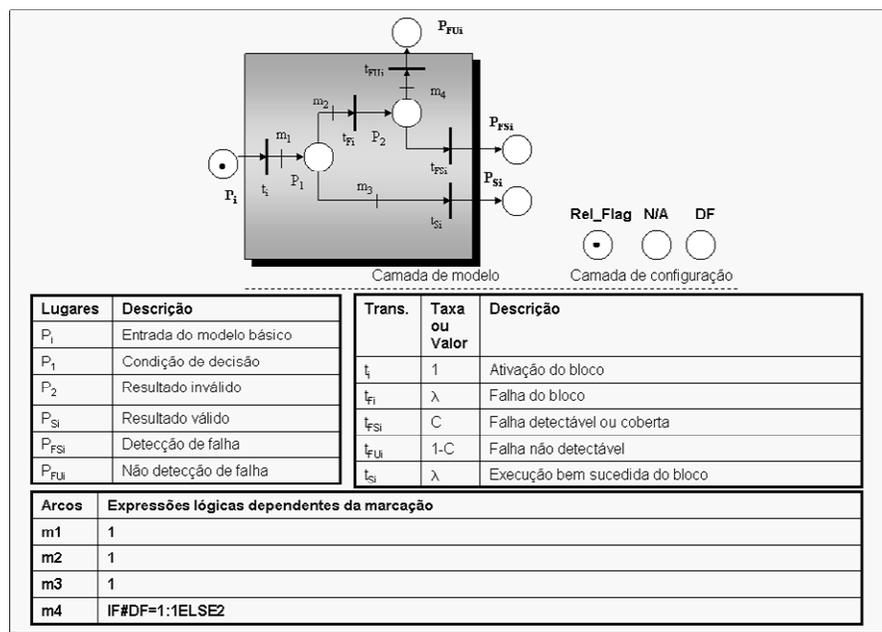


Figura 5.7 Modelo MDP do bloco básico

As expressões lógicas das transições imediatas em conflito, representadas pelas transições t_{Si} e t_{Fi} na Figura 5.7 contêm funções analíticas de confiabilidade e disponibilidade definidas no nível hierárquico 4 através do modelo EDSPN associado ao bloco básico sem replicação, enquanto que as transições t_{FUi} e t_{FSi} contêm as probabilidades de detecção ou não detecção das falhas, representadas pelo fator de cobertura C. Os *flags* ou parâmetros de controle, definidos na Tabela 5.6 e apresentados na Figura 5.7, são representados por lugares isolados, e são responsáveis pela seleção da configuração estrutural dos blocos no modelo, e o tipo de avaliação a ser executada, conforme a Tabela 5.7.

A combinação dos *flags* mostrados na Tabela 5.6 define as condições para seleção das características estruturais do bloco ou do conjunto de blocos, e o tipo da avaliação a ser executada, conforme a Tabela 5.7. As expressões condicionais da Tabela 5.8, definem os pesos das transições imediatas conflitantes no modelo para avaliações de

confiabilidade, disponibilidade e segurança, as quais são dependentes das condições dos parâmetros de controle definidos na Tabela 5.6.

Tabela 5.6 Parâmetros de configuração do modelo MDP do bloco básico

Parâmetros	Descrição dos parâmetros
Rel_Flag	Parâmetro de configuração que define a avaliação a ser executada. #Rel_Flag=1, determina estimativas de confiabilidade #Rel_Flag=0, determina estimativa de disponibilidade.
DF	Parâmetro de configuração que define a detecção ou não da falha #DF=1, determina a detecção da falha #DF=0, determina o mascaramento da falha.
N_A	Parâmetro de configuração que define o tipo das estimativas realizadas #N_A=0, define estimativas por meio de expressões analíticas #N_A=1, define estimativas por meio de expressões numéricas.
C	Fator de Cobertura que determina a probabilidade de detecção da falha dado que a falha existe, para as transições conflitantes t_{FSi} e t_{FUj} .

Tabela 5.7 Composição dos parâmetros de configuração do modelo MDP bloco básico

Param. de configuração	Descrição
#Rel_Flag=1AND#N_A=0	Estimativa de confiabilidade por meio de expressões analíticas.
#Rel_Flag=0AND#N_A=0	Estimativa de disponibilidade por meio de expressões analíticas.
#DF=0	Avaliações de confiabilidade e disponibilidade através de mascaramento de falha.
#DF=1	Avaliações de confiabilidade, disponibilidade e segurança por meio de detecção de falha (Fator de cobertura = C)

Tabela 5.8 Peso das transições imediatas em conflito no modelo MDP do bloco básico

Transições imediatas	Expressões lógicas condicionais	Taxas ou Valores
t_i	1	-
t_{Si}	IF #Rel_Flag = 1AND#N_A=0: $(\exp^{-\lambda t})$ IF #Rel_Flag = 0AND#N_A=0: $\frac{1}{1 + \frac{\lambda}{\mu}}$	λ e μ
t_{Fi}	IF#Rel_Flag=1AND#N_A=0: $(1 - \exp^{-\lambda t})$ IF#Rel_Flag=0AND#N_A=0: $1 - \left(\frac{1}{1 + \frac{\lambda}{\mu}} \right)$	λ e μ
t_{FSi}	C	C
t_{FUj}	(1-C)	(1-C)

A execução do bloco por meio do modelo MDP é iniciada pela presença de um *token* no lugar P_i . Estimativas são geradas pela presença de *token* no lugar de saída P_{Si} , estabelecendo uma condição válida, pela presença de *token* no lugar de saída P_{FSi} , definindo uma condição inválida, com detecção de falha, ou ainda pela presença de *token* no lugar de saída P_{FUi} , determinando uma condição inválida sem detecção de falha. As estimativas de dependabilidade e as respectivas métricas são definidas na Tabela 5.9.

Tabela 5.9 Estimativas x Métricas do modelo MDP do bloco básico

Estimativas	Métricas
Confiabilidade	$P\{\#P_{Si}=1\}$
Inconfiabilidade	$P\{\#P_{FSi}+\#P_{FUi}=1\}$
Falhas seguras	$P\{\#P_{FSi}=1\}$
Falhas inseguras	$P\{\#P_{FUi}=1\}$
Segurança	$P\{\#P_{Si}+\#P_{FSi}=1\}$
Insegurança	$P\{\#P_{FUi}=1\}$
Disponibilidade	$P\{\#P_{Si}=1\}$
Indisponibilidade	$P\{\#P_{FSi}+\#P_{FUi}=1\}$

5.2.2 Modelo MDP: Modelo Blocos de Subsistema

O modelo descrito na Figura 5.8 corresponde a um bloco de subsistema. O bloco de subsistema representa uma combinação de diversos blocos adjacentes e, em geral, apresenta um grau de complexidade não trivial na definição de suas funções analíticas de confiabilidade e disponibilidade, a serem determinadas por modelos EDSPN no quarto nível hierárquico. Deste modo pode-se tornar mais prático e tratável, analisar-se numericamente, ao invés de analiticamente, o modelo EDSPN correspondente ao bloco de subsistema, através de estimativas de confiabilidade, disponibilidade e segurança, obtidas nos diversos pontos de verificação. O número de pontos de verificação é dependente do grau de aproximação desejado. Com a utilização de valores numéricos diretamente no modelo MDP, torna-se desnecessária a determinação da função de distribuição estatística associada ao subsistema. A utilização da função dar-se-á diretamente, através dos valores numéricos no modelo MDP sem que haja necessidade de se obter a função analítica de distribuição, associada aos pontos de verificação, por meio de alguma ferramenta estatística. Isto permite que avaliações experimentais e avaliações baseadas em modelos possam ser representadas nos modelos MDP.

O modelo MDP do bloco de subsistema permite a obtenção de valores numéricos de estado permanente para as estimativas de disponibilidade, confiabilidade e segurança. O modelo MDP descreve a curva transiente de confiabilidade, do tempo zero até um intervalo máximo de tempo através de uma série de análises de estado permanente para um número qualquer de pontos de verificação, em intervalos regulares IT , no tempo t . Os *flags*, definidos na Tabela 5.6 e apresentados na Figura 5.8 selecionam a configuração estrutural do modelo MDP, do subsistema, e o tipo de avaliação das estimativas a serem executadas, conforme a Tabela 5.10.

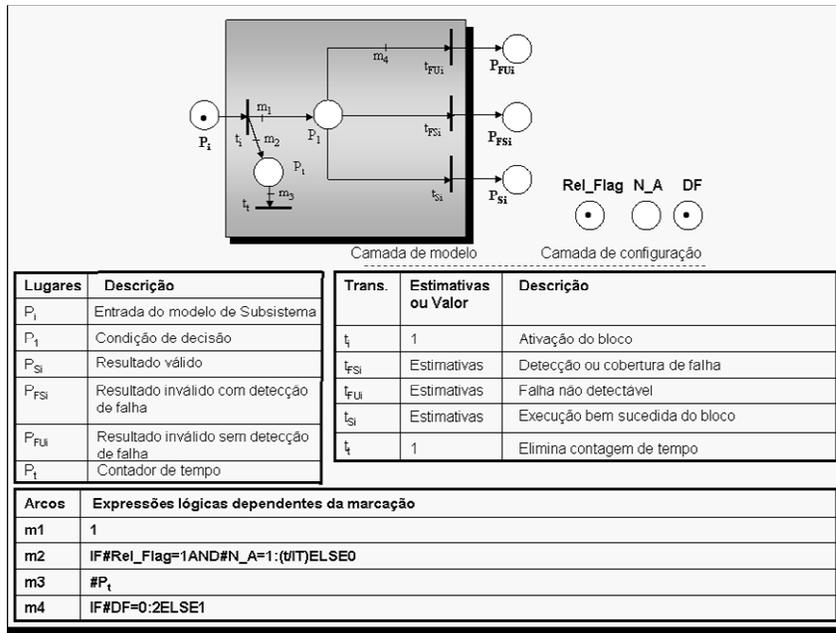


Figura 5.8 Modelo MDP para o bloco de subsistema

A multiplicidade do arco m_2 , mostrado na Figura 5.8, gerará um número de *tokens* em P_t igual a amostra de tempo que está sendo analisada. Por exemplo, se o ponto de verificação de tempo que está sendo analisado corresponde a um tempo de 100u.t., onde u.t. significa unidades de tempo, e o intervalo de tempo entre pontos de verificação, $IT=10u.t.$, logo ao analisar-se a amostra de tempo 80u.t., o número de *tokens* depositado em P_t será igual a 8, o que fará com que as expressões numéricas dos pesos das transições imediatas em conflito t_{FUi} , t_{FSi} e t_{Si} , liberem as estimativas correspondentes para falha insegura, falha segura e confiabilidade, respectivamente, obtidas do modelo EDSPN, para esse tempo.

As expressões lógicas condicionais das transições imediatas em conflito t_{Si} , t_{FSi} e t_{FUi} são apresentadas na Tabela 5.11. Para cada ponto de verificação de tempo, múltiplo de IT , num intervalo de tempo t , são definidas estimativas numéricas associadas à confiabilidade, falha segura, falha insegura, e disponibilidade, obtidas do modelo EDSPN, correspondente ao bloco de subsistema.

Tabela 5.10 Composição dos parâmetros de controle do modelo MDP de subsistema

Param. de configuração	Descrição
$\#Rel_Flag=1 \text{ AND } \#N_A=1$	Estimativa de confiabilidade por meio de valores numéricos
$\#Rel_Flag=0 \text{ AND } \#N_A=1$	Estimativa de disponibilidade por meio de valores numéricos
$\#DF=0$	Avaliações de confiabilidade e disponibilidade através de mascaramento de falha
$\#DF=1$	Avaliações de confiabilidade, disponibilidade e segurança por meio de detecção de falha (Fator de cobertura = C)

Tabela 5.11 Pesos das transições do modelo MDP de bloco de subsistema

Tran.	Expressão condicional dos pesos	Expressão de guarda	Prioridade
t_i	1	--	1
t_{Si}	IF#Rel_Flag=0AND#N_A=1(Est_Num_Dis) IF#N_A=1AND#P _t =1:(Est_Num_Conf1) IF#N_A=1AND#P _t =2:(Est_Num_Conf2).....IF#N_A=1AND #P _t =n:(Est_Num_Confn)	--	1
t_{FSi}	IF#Rel_Flag=0AND#N_A=1:(Est_Indisp_Falha_Segura)IF#N_A=1AND#P _t =1:(Est_FSeg1) IF#N_A=1AND#P _t =2:(Est_FSeg2).....IF#N_A=1AND#P _t =n:(Est_FSegn)	--	1
t_{FUi}	IF#Rel_Flag=0AND#N_A=1:(Est_Indisp_Falha_Insegura)IF#N_A=1AND#P _t =1:(Est_FISeg1) IF#N_A=1AND#P _t =2:(Est_FISeg2).....IF#N_A=1AND#P _t =n:(Est_FISegn)	--	1
t_t	#P _t	#P ₁ =0AND #P _t >0	2

Na Tabela 5.11 e nas demais tabelas, Est, Num, Disp, Conf, FSeg e FISeg são as abreviações de Estimativa, Numérica, Disponibilidade, Confiabilidade, Falha Segura e Falha Insegura, e n é o número total de pontos de verificação. O símbolo -- indica a inexistência de expressão de guarda. A definição das métricas de dependabilidade do bloco de subsistema é mostrada na Tabela 5.12

Tabela 5.12 Atributos x Métricas do bloco subsistema

Atributos	Métricas
Confiabilidade	$P\{\#P_{Si}=1\}$
Inconfiabilidade	$P\{\#P_{FSi}+\#P_{FUi}=1\}$
Falhas seguras	$P\{\#P_{FSi}=1\}$
Falhas inseguras	$P\{\#P_{FUi}=1\}$
Segurança	$P\{\#P_{Si}+\#P_{FSi}=1\}$
Insegurança	$P\{\#P_{FUi}=1\}$
Disponibilidade	$P\{\#P_{Si}=1\}$
Indisponibilidade	$P\{\#P_{FSi}+\#P_{FUi}=1\}$

5.2.3 Modelo MDP: Modelo Bloco Standby

Neste modelo são apresentadas as diversas formas de replicação para o bloco básico, composto por distribuições exponenciais para falha e reparo. O modelo MDP do bloco *standby* é representado por funções analíticas correspondentes às replicações passivas dos tipos *coldstandby* e *warmstandby*, e à replicação semi-ativa do tipo *hotstandby*, as quais foram definidas no Capítulo 4. Neste modelo considera-se a comutação do bloco ativo para o bloco passivo como sendo livre de falhas. Os modelos

correspondentes às diversas réplicas podem assumir diferentes taxas de falha quando nos modos operacional e em espera (*standby*):

- quando a réplica primária, ou qualquer uma das réplicas que a substitua, apresentar uma taxa de falha λ , e as réplicas passivas em espera forem livres de falhas, o modelo a ser considerado é o *coldstandby* e a função de confiabilidade do modelo corresponde a uma distribuição de Erlang;
- quando as taxas de falha ξ das réplicas passivas em espera forem diferentes da taxa de falha λ das réplicas quando na condição ativa, considera-se o modelo *warmstandby*, e a função de confiabilidade corresponde a uma distribuição hipoexponencial;
- quando todas as réplicas estiverem ativas, embora só a réplica primária tenha a capacidade de fornecer um resultado, e as taxas de falha das réplicas forem as mesmas e iguais a λ , considera-se o modelo *hotstandby* e a função de confiabilidade correspondente é uma distribuição exponencial cuja taxa de falha é dada pela expressão:

$$\lambda_{HS} = \frac{\lambda}{\sum_{i=1}^n \frac{1}{i}} \quad (5.1)$$

O modelo MDP do bloco *standby*, assim como os demais modelos MDP, é constituído pelas camadas de modelo e de configuração podendo ser implementado de duas maneiras distintas, considerando-se estimativas analíticas e numéricas:

- o primeiro modo faz uso das funções analíticas para confiabilidade e disponibilidade mostradas nas tabelas 5.2, 5.3 e 5.4 e desenvolvidas no nível hierárquico 4 através de modelos EDSPN associados as diversas formas de replicação. Este modelo apresenta uma dificuldade: como a densidade de probabilidade de falha do bloco *standby* é uma convolução das densidades individuais das réplicas ativa e passivas, o fator de cobertura apresentado como uma constante no modelo EDSPN deve ser transformado para que possa se manter válido. O novo fator de cobertura C_1 utilizado no modelo MDP do bloco *standby* é a relação entre a probabilidade de falha segura pela estimativa de inconfiabilidade do bloco, enquanto o complementar do fator de cobertura $1 - C_1$ é a relação entre a probabilidade de falha insegura pela estimativa de inconfiabilidade do bloco. Estes valores, C_1 e $1-C_1$, resultantes da transformação do fator de cobertura C e $1-C$, são obtidos do modelo EDSPN correspondente ao bloco *standby*. Caso se proceda a uma análise transiente do modelo EDSPN para diversos intervalos de tempo, pode-se definir uma função polinomial em função do fator de cobertura original C e colocar nas expressões das transições imediatas conflitantes t_{FU_i} e t_{FS_i} . Este modelo analítico apresenta como vantagem, a flexibilidade de utilização do mesmo modelo para diversos valores de cobertura. Na Figura 5.9 são mostrados o modelo e as expressões de multiplicidade dos arcos a serem utilizadas.

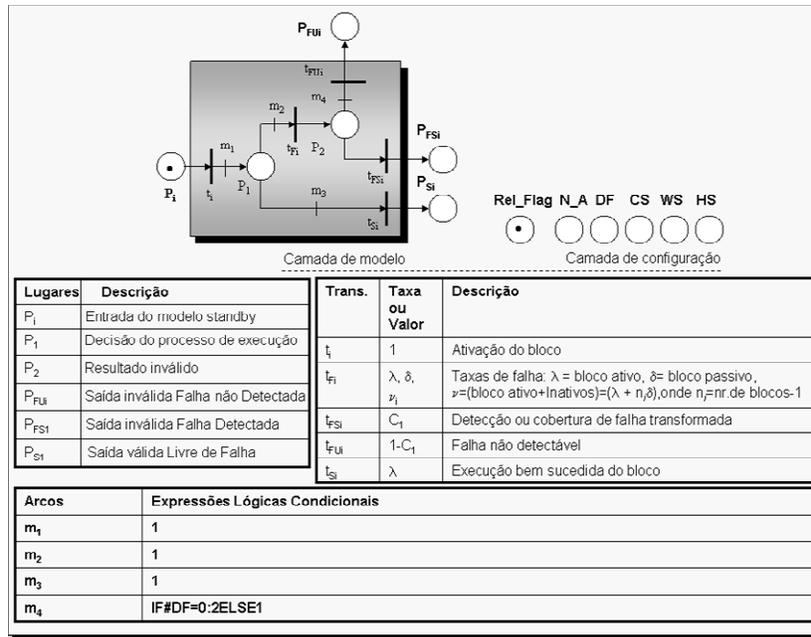


Figura 5.9 Modelo MDP do bloco standby com estimativas analíticas

- um segundo modo faz uso de estimativas numéricas para confiabilidade, disponibilidade, falha segura e falha insegura, obtidas no nível hierárquico 4 através dos modelos EDSPN, correspondentes as diferentes formas de replicação e número de réplicas. Este modelo é mostrado na Figura 5.10. O modelo de bloco *standby* com estimativas numéricas é bastante similar ao modelo de bloco de subsistema, exceto pelos parâmetros de controle CS, WS e HS responsáveis pela seleção do tipo de replicação adotada. Tanto os valores de confiabilidade quanto os valores de falha segura, falha insegura e disponibilidade são gerados, os quais são necessários para determinação das estimativas do modelo a serem utilizados no diagrama do sistema como um todo.

Tabela 5.13 Definição dos parâmetros de controle do modelo de bloco standby

Parâmetros	Descrição dos parâmetros
Rel_Flag	Parâmetro de configuração que define a avaliação a ser executada. Estimativas de confiabilidade e segurança são geradas quando #Rel_Flag=1, enquanto que estimativas de disponibilidade são geradas quando #Rel_Flag=0.
DF	Parâmetro de configuração responsável pelo mascaramento da falha quando #DF=0 ou pela detecção da falha quando #DF=1.
N_A	Parâmetro de configuração que habilita expressões analíticas, quando #N_A=0 ou expressões numéricas quando #N_A=1, para as transições conflitantes t_{Si} e t_{Fi} .
C	Fator de Cobertura que determina a probabilidade de detecção da falha, dado que a falha exista, para as transições conflitantes t_{FSi} e t_{FUi} .
CS	Parâmetro que habilita a técnica de replicação do tipo <i>coldstandby</i> .
WS	Parâmetro que habilita a técnica de replicação do tipo <i>warmstandby</i> .
HS	Parâmetro que habilita a técnica de replicação do tipo <i>hotstandby</i> .

A escolha do tipo de análise, do requisito de dependabilidade e da expressão analítica ou numérica a ser utilizada, é definida por meio dos parâmetros de configuração através da Tabela 5.13. Como se pode observar na Tabela 5.14, dependendo da combinação dos diversos parâmetros de configuração obtêm-se uma série de estimativas para as diferentes técnicas de replicação associadas ao bloco primário.

Tabela 5. 14 Composição dos parâmetros de controle do modelo MDP de bloco *standby*

Param. de configuração	Descrição
#Rel_Flag=1,#XS=1, Demais <i>flags</i> = 0	Avaliações analíticas de confiabilidade para os modelos <i>standby</i> , onde X= C (<i>ColdStandby</i>) ou = W (<i>WarmStandby</i>) ou = H (<i>HotStandby</i>) Com mascaramento de falha.
#Rel_Flag=1, #XS=1, #N_A=1, Demais <i>flags</i> =0	Estimativas numéricas de confiabilidade para os modelos <i>standby</i> com mascaramento de falha.
#Rel_Flag=1,#XS=1, #DF=1, Demais <i>flags</i> = 0	Estimativas analíticas de confiabilidade e segurança para os modelos <i>standby</i> (com fator de cobertura = C)
#Rel_Flag=1,#XS=1,#DF=1, #N_A=1, Demais <i>flags</i> =0	Estimativas numéricas de confiabilidade e segurança para os modelos <i>standby</i> (com fator de cobertura = C)
#Rel_Flag=0,#XS=1, Demais <i>flags</i> = 0	Estimativa analítica de disponibilidade para os modelos <i>standby</i> com mascaramento de falha.
#Rel_Flag=0, #XS=1, #N_A=1, Demais <i>flags</i> = 0	Estimativa numérica de disponibilidade para os modelos <i>standby</i> com mascaramento de falha.
#Rel_Flag=0,#XS=1, #DF=1 Demais <i>flags</i> = 0	Estimativa analítica de disponibilidade para os modelos <i>standby</i> (com fator de cobertura = C)
#Rel_Flag=0,#XS=1,#DF=1 #N_A=1, Demais <i>flags</i> = 0	Estimativa numérica de disponibilidade para os modelos <i>standby</i> (com fator de cobertura = C)

Quando aspectos de detecção de falha não são relevantes, como no caso do mascaramento de falhas, inibi-se o disparo da transição t_{FU_i} através do peso do arco de entrada m_4 . Neste caso, obtêm-se apenas as estimativas de confiabilidade e de disponibilidade, uma vez que as estimativas de segurança estão associadas aos fatores de cobertura. A definição de métricas para as estimativas de dependabilidade dos blocos *standby* é semelhante aquelas do modelo de bloco de subsistema e pode ser vista na Tabela 5.15.

Tabela 5.15 Atributos x Métricas bloco *standby*

Atributos	Métricas
Confiabilidade	$P\{\#P_{Si}=1\}$
Inconfiabilidade	$P\{\#P_{FSi}+\#P_{FU_i}=1\}$
Falhas seguras	$P\{\#P_{FSi}=1\}$
Falhas inseguras	$P\{\#P_{FU_i}=1\}$
Segurança	$P\{\#P_{Si}+\#P_{FSi}=1\}$
Insegurança	$P\{\#P_{FU_i}=1\}$
Disponibilidade	$P\{\#P_{Si}=1\}$
Indisponibilidade	$P\{\#P_{FSi}+\#P_{FU_i}=1\}$

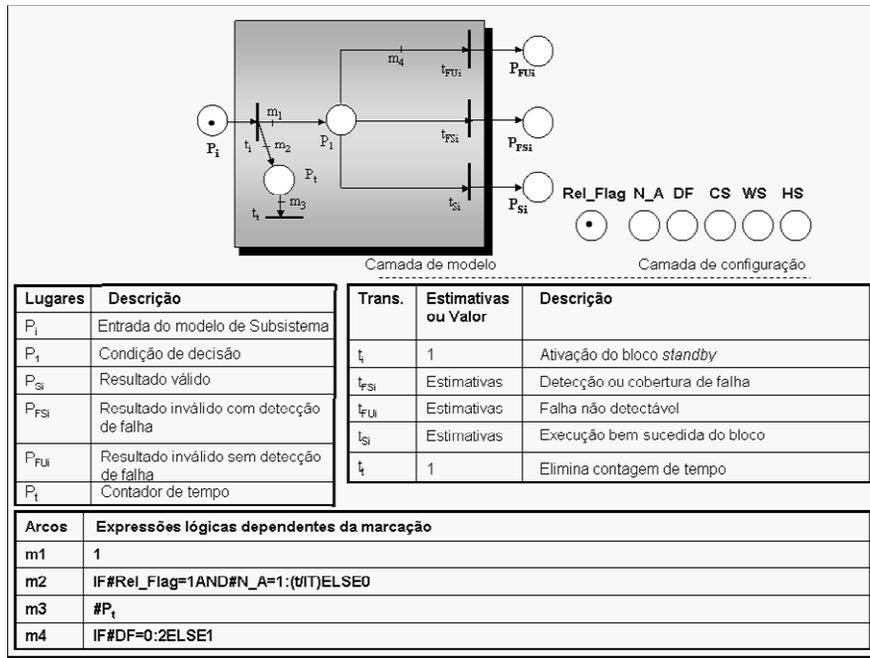


Figura 5.10 Modelo MDP do bloco *standby* com estimativas numéricas

5.2.4 Modelo MDP: Modelo Bloco de Decisão Básico

O modelo de bloco de decisão básico, de fundamental importância nos diagramas de sistema dependáveis e parametrizados, permite a definição de diversas configurações entre os vários modelos de blocos descritos. Os modelos MDP dos blocos de decisão, assim como os modelos dos demais blocos dependáveis e parametrizados, são compostos por uma camada de modelo e uma camada de configuração. A camada de modelo é formada pelos elementos da rede de Petri associados ao modelo MDP do bloco de decisão, enquanto a camada de configuração é constituída por *flags*, responsáveis pela configuração estrutural dos modelos MD correspondente aos blocos conectados. Os *flags* utilizados nos modelos dos blocos de decisão, bem como suas respectivas descrições, são mostrados na Tabela 5.16.

As configurações entre blocos aqui consideradas, são em geral do tipo serial, paralelo e *m/n*. A configuração *m/n*, como já descrito anteriormente, é um caso especial de paralelismo na qual o sistema para funcionar convenientemente necessita que pelo menos *m-n* blocos em paralelo estejam funcionando convenientemente, onde *m* é o número de blocos em falha. Também são consideradas configurações especiais de paralelismo entre blocos como a configuração NMR, caso especial da configuração *m/n*, na qual *n* é um número inteiro e ímpar, e a configuração *flux-summing* um caso especial de paralelismo utilizado em sistemas de aviação, que produz um resultado único por meio de blocos redundantes e compensação de fluxo, num sistema de controle de realimentação fechada [70]. A técnica de *flux-summing* não corresponde a um processo de votação, mas tem o mesmo efeito de mascaramento de falhas, quando da compensação de fluxo, devido ao fluxo que deixou de ser produzido por um determinado bloco em falha. Não apenas as conexões anteriormente citadas como também conexões

série/paralela complexas solucionadas pela aplicação do teorema das probabilidades totais em [136] ou ainda conexões não-série/não-paralela como descrita em [126] são resolvidas pelos modelos MDP por meio dos blocos de decisão. O modelo MDP do bloco de decisão, mostrado na Figura 5.11, é utilizado para blocos concorrentes e para blocos seriais. As multiplicidades dos arcos do bloco de decisão em função dos parâmetros de configuração, definidos na Tabela 5.16, são mostrados nas Tabelas 5.17, 5.18 e 5.19. A Tabela 5.17 mostra a multiplicidade dos arcos de entrada da transição t_{FS} , correspondente a falha segura; A Tabela 5.18 mostra a multiplicidade dos arcos de entrada da transição t_{FU} , correspondente a falha insegura; A Tabela 5.19 mostra a multiplicidade dos arcos de entrada da transição t_S correspondente a sucesso.

Tabela 5.16 Parâmetros de configuração do modelo MDP de bloco de decisão

Parâmetros	Descrição dos parâmetros
Rel_Flag	Parâmetro de configuração que define o tipo de avaliação a ser executada: #Rel_Flag=0 → Estimativas de disponibilidade #Rel_Flag=1 → Estimativas de confiabilidade e segurança.
DF	Parâmetro de configuração de detecção de falhas: #DF=0 → Mascaramento de falhas #DF=1 → Detecção de falhas por fator de cobertura C.
N_A	Parâmetro de configuração que define o tipo de expressão das transições conflitantes t_{Si} e t_{Fi} : #N_A=0 → Expressões analíticas #N_A=1 → Expressões numéricas
Ser_Par	Configuração dos blocos concorrentes: #Ser_Par=1 → Configuração Serial #Ser_Par=0 → Configuração Paralela
M_N	Configuração m/n a qual libera um resultado válido quando pelo menos $n-m$ blocos estão funcionando convenientemente (confiavelmente), onde m é o número de blocos em falha.
NMR	Configuração de redundância modular N, a qual apresenta um resultado válido quando pelo menos $(n+1)/2$ blocos estão funcionando convenientemente e n é o número total de blocos.
FS	Configuração <i>Flux-Summing</i> , a qual apresenta um resultado válido enquanto existir pelo menos 1 bloco fornecendo o fluxo necessário e os demais blocos tenham tido seus fluxos compensados por um processo de <i>loop</i> fechado.

Exemplo de aplicação do bloco de decisão: Deseja-se verificar se a saída de 3 blocos básicos, configurados segundo um mecanismo tolerante a falhas, TMR, libera um resultado de sucesso, de falha segura ou falha insegura. Inicialmente, o bloco de decisão recebe os resultados dos três blocos, através das entradas P_{UDi} , P_{FDi} e P_{SDi} a ele ligados. Logo a quantidade de tokens nas entradas do bloco de decisão $\#P_{UDi} + \#P_{FDi} + \#P_{SDi} = 3$, onde o símbolo # significa quantidade de tokens no lugar definido. Cada um dos blocos pode estar em uma condição operacional ou falha. Há, portanto, oito combinações de resultados possíveis. Como a configuração é TMR, o flag #NMR deve estar ativo, ou

seja, #NMR=1 e os demais inativos. Supondo detecção de falhas perfeita, tem-se #DF=0. Em qualquer instante apenas uma das três saídas do bloco de decisão estará ativa. Quando o resultado do bloco de decisão é válido, a saída D_{Si} fica ativa, enquanto as saídas de falha segura, D_{FSi} , e de falha insegura D_{FU_i} ficam inativas. Para que a saída do bloco de decisão seja válida é necessário que a saída da configuração TMR resulte válida (dois ou mais blocos de entrada sejam válidos), ou seja, para que haja um token em D_{Si} é necessário que a transição t_{Si} seja habilitada e dispare.

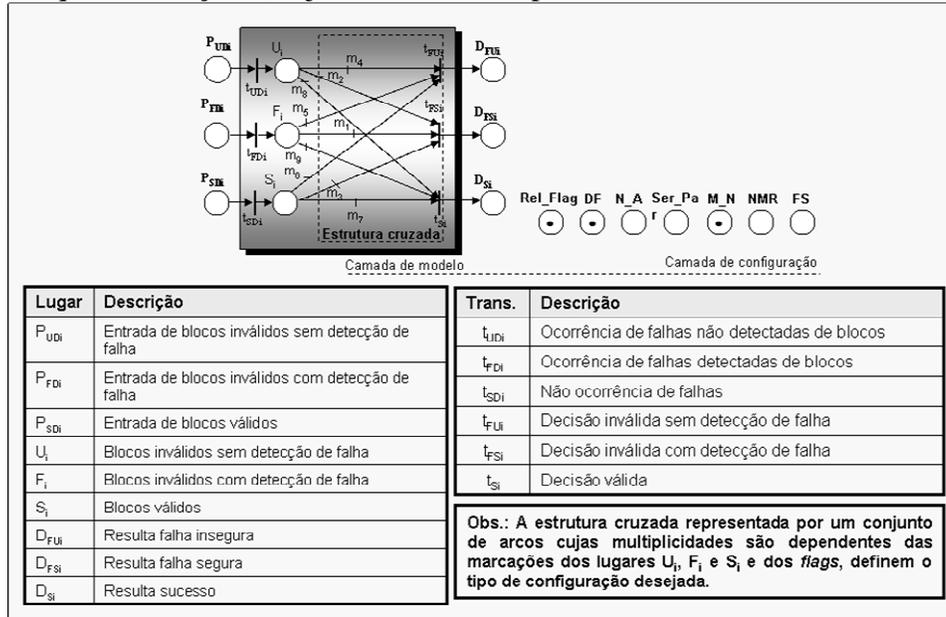


Figura 5.11 Modelo MDP do bloco de decisão

Isto ocorre numa configuração TMR, quando os resultados da Tabela 5.19 permitem habilitar a transição t_{Si} . Retirando da Tabela 5.19 apenas a expressão da configuração TMR, isto é, NMR para $n_1=3$, tem-se:

$$m_7 : \dots \text{IF}\#NMR1=1\text{AND}\#M_N1=0\text{AND}(\#S1>((n1-1)/2)): \#S1 \quad (5.2)$$

$$m_8 : \dots \text{IF}\#NMR1=1\text{AND}\#M_N1=0:0 \quad (5.3)$$

$$m_9 : \dots \text{IF}\#NMR1=1\text{AND}\#M_N1=0\text{AND}(\#S1>((n1-1)/2)): (n1-\#S1) \quad (5.4)$$

A expressão em m_7 , diz que, quando o número de tokens no lugar S_i for maior do que $(3-1)/2$, ou seja $\#S_i > 1$, a transição t_{Si} estará habilitada para este arco de entrada. $\#S_i > 1$ indica que existem 2 ou 3 blocos de entrada válidos; a expressão m_8 , significa que a detecção sendo perfeita, o lugar $\#U_i$ não terá token, uma vez que não há blocos com falha não detectável. Logo o número de tokens em $U_i = 0$; a expressão m_9 significa que o número de blocos inválidos, porém com falha detectável, deve ser $(n_1 - S_i) = (3 - S_i) < 2$, ou seja, 0 ou 1.

Caso o número de tokens em $\#S_i$, $\#U_i$ e $\#F_i$ satisfaçam as condições estabelecidas pelos arcos m_7 , m_8 e m_9 , respectivamente, a transição S_i , dispara colocando um token na saída D_{Si} , indicando que a saída do bloco de decisão para três blocos básicos configurados segundo um mecanismo de tolerância a falhas do tipo TMR, é uma saída válida. As expressões para as saídas D_{FS_i} e D_{FU_i} fornecem um resultado zero. A estrutura cruzada do bloco de decisão executa todas as oito possíveis combinações dos blocos de entrada de uma configuração TMR. Dependendo da

quantidade de tokens em cada uma das entradas da estrutura cruzada, U_i , F_i e S_i , e da regra do mecanismo TMR, três resultados são possíveis: sucesso, falha detectável e falha não detectável.

Tabela 5.17 Multiplicidade dos arcos de entrada de uma transição de falha segura

Arco	Expressões Lógicas Condicionais da Multiplicidade dos Arcos
m_1	$IF\#N_A1=1:1IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1>\#U1)AND(\#S1+\#F1+\#U1=n1):\#F1$ $IF\#NMR1=1AND\#M_N1=0AND(\#F1>((n1-1)/2)):\#F1$ $IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#F1+\#S1=n1)AND\#F1>0:\#F1$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#F1=n1:n1$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND(\#F1=n1):\#F1ELSE(n1+1)$
m_2	$IF\#N_A1=1:0IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1>\#U1)AND(\#S1+\#F1+\#U1=n1):\#U1$ $IF\#NMR1=1AND\#M_N1=0:0$ $IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#F1+\#S1=n1):0$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#F1=n1:0$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND(\#F1=n1):0ELSE(n1+1)$
m_3	$IF\#N_A1=1:0IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1>\#U1)AND(\#S1+\#F1+\#U1=n1):\#S1$ $IF\#NMR1=1AND\#M_N1=0AND(\#F1>((n1-1)/2)):(n1-\#F1)$ $IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#F1+\#S1=n1):(n1-\#F1)$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#F1=n1:0$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND\#F1=n1:0ELSE(n1+1)$

Tabela 5.18 Multiplicidade dos arcos de entrada de uma transição de falha insegura

Arco	Expressões Lógicas Condicionais da Multiplicidade dos Arcos
m_4	$IF\#N_A1=1:1IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1<=\#U1)AND(\#S1+\#F1+\#U1=n1):\#U1$ $IF\#NMR1=1AND\#U1>0AND\#M_N1=0:0$ $IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#U1>0):\#U1$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#U1>0AND(\#U1+\#F1=n1):\#U1$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND(\#U1+\#F1+\#S1=n1)AND((\#U1>\#S1)OR(\#U1=\#S1)AND(\#F1<n1)):\#U1ELSE(n1+1)$
m_5	$IF\#N_A1=1:0IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1<=\#U1)AND(\#S1+\#F1+\#U1=n1):\#F1$ $IF\#NMR1=1AND\#M_N1=0:0$ $IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#U1>0):(n1-\#S1-\#U1)$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#U1>0AND(\#U1+\#F1=n1):(n1-\#U1)$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND(\#U1+\#F1+\#S1=n1)AND((\#U1>\#S1)OR(\#U1=\#S1)AND(\#F1<n1)):(n1-\#S1-\#U1)ELSE(n1+1)$
m_6	$IF\#N_A1=1:0IF\#M_N1=1AND(\#F1+\#U1>=q1)AND(\#F1<=\#U1)AND(\#S1+\#F1+\#U1=n1):\#S1$ $IF\#NMR1=1AND\#M_N1=0:0IF\#Ser_Par1=1AND(\#NMR1+\#M_N1+\#FS1)=0AND(\#U1+\#F1>0):(n1-\#U1-\#F1)$ $IF(\#Ser_Par1+\#M_N1+\#NMR1+\#FS1)=0AND\#U1>0AND(\#U1+\#F1=n1):0$ $IF(\#NMR1+\#M_N1)=0AND\#FS1=1AND(\#U1+\#F1+\#S1=n1)AND((\#U1>\#S1)OR(\#U1=\#S1)AND(\#F1<n1)):(n1-\#U1-\#F1)ELSE(n1+1)$

Tabela 5.19 Multiplicidade dos arcos de entrada de uma transição de sucesso

Arco	Expressões Lógicas Condicionais da Multiplicidade dos Arcos
m ₇	IF#N_A1=1:1IF#M_N1=1AND(#F1+#U1<q1)AND(#F1+#U1+#S1=n1):#S1 IF#NMR1=1AND#M_N1=0AND(#S1>((n1-1)/2)):#S1 IF#Ser_Par1=1AND(#NMR1+#M_N1+#FS1)=0:n1 IF(#Ser_Par1+#M_N1+#NMR1+#FS1)=0AND#S1>0:#S1 IF(#NMR1+#M_N1)=0AND#FS1=1AND((#S1+#F1=n1)AND(#S1>0))OR (#S1>#F1+#U1)OR((#S1>#U1)AND(#U1=1)):#S1ELSE(n1+1)
m ₈	IF#N_A1=1:0IF#M_N1=1AND(#F1+#U1<q1)AND(#F1+#U1+#S1=n1):#U1 IF#NMR1=1AND#M_N1=0:0IF#Ser_Par1=1AND(#NMR1+#M_N1+#FS1)=0:0 IF(#Ser_Par1+#M_N1+#NMR1+#FS1)=0:(n1-#S1-#F1) IF(#NMR1+#M_N1)=0AND#FS1=1AND((#S1+#F1=n1)AND(#S1>0))OR(#S1>#F1 +#U1)OR((#S1>#U1)AND(#U1=1)):(n1-#S1-#F1)ELSE(n1+1)
m ₉	IF#N_A1=1:0IF#M_N1=1AND(#F1+#U1<q1)AND(#F1+#U1+#S1=n1):#F1 IF#NMR1=1AND#M_N1=0AND(#S1>((n1-1)/2)):(n1#S1) IF#Ser_Par1=1AND(#NMR1+#M_N1+#FS1)=0:0 IF(#Ser_Par1+#M_N1+#NMR1+#FS1)=0:(n1-#S1-#U1) IF(#NMR1+#M_N1)=0AND#FS1=1AND((#S1+#F1=n1)AND(#S1>0))OR(#S1>#F1 +#U1)OR((#S1>#U1)AND(#U1=1)):(n1-#S1-#U1)ELSE(n1+1)

Dependendo da combinação dos *flags* na camada de configuração, diversas configurações e análises podem ser definidas. As decisões a serem tomadas em função das diversas configurações estão definidas na Tabela 5.20, enquanto que as métricas a serem utilizadas para se auferir as estimativas de saída do bloco de decisão são descritas na Tabela 5.21.

O parâmetro de configuração N_A, responsável pela habilitação de expressões analíticas ou numéricas para cada bloco, é relevante no modelo de bloco de decisão, quando o modelo do bloco de decisão está concatenado a um ou mais modelos de blocos básicos, e a sua entrada depende das transições analíticas ou numéricas dos blocos básicos em questão. A composição do modelo de decisão, na configuração de blocos seriais, com o modelo de blocos básicos, forma os modelos de blocos seriais múltiplos, a serem descritos na próxima seção. As diversas estimativas produzidas pelos blocos de decisão refletem os dados fornecidos pelos blocos de entrada e a configuração a que eles estão submetidos.

Tabela 5.20 Composição dos parâmetros de controle do modelo de bloco de decisão

Flags de controle	Descrição
#Rel_Flag=1, Demais <i>flags</i> = 0	Estimativas de confiabilidade para uma configuração de blocos em paralelo com mascaramento de falhas
#Rel_Flag=1,#X_Flag=1, Demais <i>flags</i> =0	Avaliação exclusiva de confiabilidade para cada uma das configurações com mascaramento: X_Flag =M_N → Configuração m/n = NMR → Redundância Modular N = FS → <i>Flux-Summing</i> =Ser_Par → Serial (não utilizada quando os blocos são concorrentes)
#Rel_Flag=1,#DF=1, Demais <i>flags</i> = 0	Avaliação de confiabilidade para uma configuração de blocos em paralelo com fator de cobertura = C.
#Rel_Flag=1,#X_Flag=1, #DF=1, Demais <i>flags</i> = 0	Avaliações exclusivas de confiabilidade para cada uma das configurações com fator de cobertura = C: X_Flag =M_N → Configuração m/n = NMR → Redundância Modular N = FS → <i>Flux-Summing</i> =Ser_Par → Serial (não utilizado para blocos concorrentes)
#Rel_Flag=0, Demais <i>flags</i> = 0	Avaliação de disponibilidade em uma configuração de blocos paralelos com mascaramento
#Rel_Flag=0, #X_Flag=1, Demais <i>flags</i> = 0	Avaliação exclusiva de disponibilidade para cada uma das configurações com mascaramento X_Flag =M_N → Configuração m/n = NMR → Redundância Modular N = FS → <i>Flux-Summing</i> =Ser_Par → Serial (não utilizado para blocos concorrentes)
#Rel_Flag=0, #DF=1 Demais <i>flags</i> = 0	Avaliação de disponibilidade para uma configuração de blocos em paralelo com fator de cobertura C
#Rel_Flag=0, #X_Flag=1, #DF=1, Demais <i>Flags</i> = 0	Avaliação exclusiva de disponibilidade para cada uma das configurações, com fator de cobertura C: X_Flag =M_N → Configuração m/n = NMR → Redundância Modular N = FS → <i>Flux-Summing</i> =Ser_Par → Serial (não utilizado para blocos concorrentes)

Tabela 5.21 Atributos x Métricas do bloco de decisão

Atributos	Métricas
Confiabilidade	$P\{\#D_{Si}=1\}$
Inconfiabilidade	$P\{\#D_{FSi}+\#D_{FUi}=1\}$
Falhas seguras	$P\{\#D_{FSi}=1\}$
Falhas inseguras	$P\{\#D_{FUi}=1\}$
Segurança	$P\{\#D_{Si}+\#D_{FSi}=1\}$
Insegurança	$P\{\#D_{FUi}=1\}$
Disponibilidade	$P\{\#D_{Si}=1\}$
Indisponibilidade	$P\{\#D_{FSi}+\#D_{FUi}=1\}$

5.2.5 Modelo MDP: Modelo Bloco Serial Múltiplo

Este modelo é usado para minimizar as dimensões e a complexidade dos diagramas de sistemas dependáveis e parametrizados, pela transformação de um conjunto de blocos em série em um único bloco composto. O modelo de bloco serial múltiplo é composto por modelos de blocos em série concatenados a um bloco de decisão, conforme mostrado na Figura 5.12. A multiplicidade dos arcos, mostrada na Tabela 5.22, determina as regras de decisão para n blocos em série, através de estimativas analíticas ou numéricas dos blocos, com as mesmas, ou diferentes, taxas de falha e de reparo. Pode-se definir as regras de configuração serial para um bloco serial múltiplo atribuindo-se ao *flag* #Ser_Par o valor 1 nas Tabelas 5.17, 5.18 e 5.19. A Tabela 5.22 é um caso particular das Tabelas 5.17, 5.18 e 5.19.

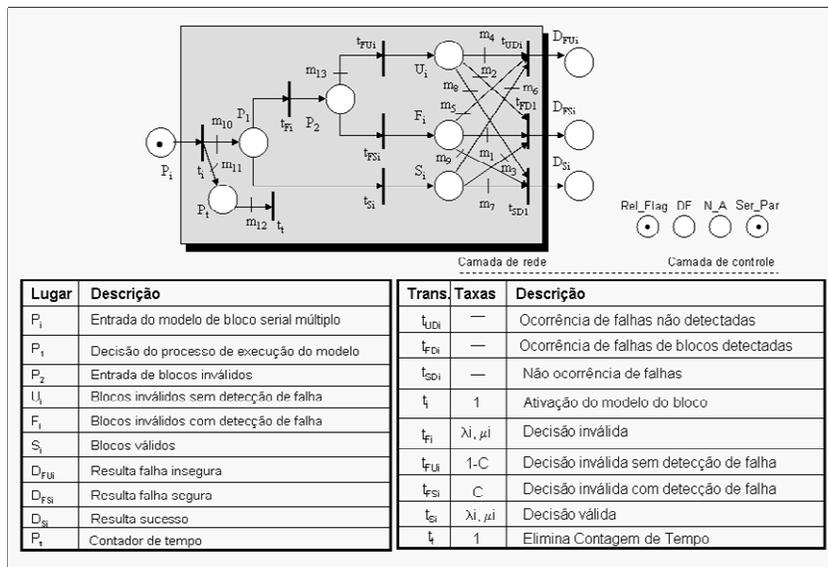


Figura 5.12 Modelo MDP dos blocos seriais múltiplos

O número de blocos em série é representado pelo número n de *tokens*, presentes em P_1 , após disparo do arco m_{10} o qual apresenta uma expressão de multiplicidade de arco igual a n . Desse modo, reduz-se o número de lugares, transições e arcos correspondentes aos n blocos em série, associando-se a cada bloco um *token* [3] e

utilizando-se as mesmas camadas de modelo e de configuração comuns a todos. De fato n componentes, não necessariamente idênticos, podem ser modelados por um único modelo com n tokens no lugar inicial. Os modelos dos blocos considerados podem ser de qualquer tipo, desde que apresentem expressões analíticas para confiabilidade e disponibilidade ou expressões numéricas definidas através dos modelos EDSPN do nível hierárquico 4. Diferentes taxas de falha poderão ser atribuídas aos blocos nas expressões analíticas dos atributos de dependabilidade. O número de blocos n e os parâmetros de configuração das respectivas expressões lógicas poderão ser modificados de modo a satisfazer restrições impostas pelos usuários ao sistema.

Tabela 5.22 Multiplicidade dos arcos do modelo MDP dos blocos serias múltiplos

Arcos	Expressões Lógicas Condicionais da multiplicidade dos arcos
m ₁	IF#Ser_Par ₁ =1AND(#F ₁ +#S ₁ =n ₁)AND#F ₁ >0:#F ₁ ELSE(n ₁ +1)
m ₂	IF#Ser_Par ₁ =1AND(#F ₁ +#S ₁ =n ₁):0ELSE(n ₁ +1);
m ₃	IF#Ser_Par ₁ =1AND(#F ₁ +#S ₁ =n ₁):(n ₁ -#F ₁) ELSE(n ₁ +1)
m ₄	IF#Ser_Par ₁ =1AND(#U ₁ >0):#U ₁ ELSE(n ₁ +1)
m ₅	IF#Ser_Par ₁ =1AND(#U ₁ >0):(n ₁ -#S ₁ -#U ₁) ELSE(n ₁ +1)
m ₆	IF#Ser_Par ₁ =1AND#NMR ₁ =0AND(#U ₁ +#F ₁ >0):(n ₁ -#U ₁ -#F ₁) ELSE(n ₁ +1)
m ₇	IF#Ser_Par ₁ =1:n ₁ ELSE(n ₁ +1)
m ₈	IF#Ser_Par ₁ =1:0ELSE(n ₁ +1)
m ₉	IF#Ser_Par ₁ =1:0ELSE(n ₁ +1)
m ₁₀	n ₁
m ₁₁	IF#Rel_Flag=1AND#N_A=1:(t/IT)ELSE0
m ₁₂	#P _t
m ₁₃	IF#DF=0:(n ₁ +1)ELSE1

Tabela 5.23 Composição dos parâmetros de configuração do modelo bloco serial múltiplo

Parâmetros	Descrição dos parâmetros
Rel_Flag	Parâmetro de configuração que define o tipo de avaliação a ser executada: #Rel_Flag=0 → Estimativas de disponibilidade #Rel_Flag=1 → Estimativas de confiabilidade e segurança
DF	Parâmetro de configuração de detecção de falhas: #DF=0 → Mascaramento de falhas #DF=1 → Detecção de falhas por fator de cobertura C
N_A	Parâmetro de configuração que define o tipo de expressão das transições conflitantes t _{Si} e t _{Fi} : #N_A=0 → Expressões analíticas #N_A=1 → Expressões numéricas
Ser_Par	Configuração dos blocos concorrentes: #Ser_Par=1 → Configuração Serial #Ser_Par=0 → Configuração Paralela (não é utilizada nas configurações seriais) Obs.: neste modelo o valor do parâmetro Ser_Par é sempre 1.

Tabela 5.24 Pesos das transições no modelo MDP do bloco serial múltiplo

Tran.	Expressões lógicas condicionais	Guarda	P
t_i	1	---	1
t_{si}	IF#Rel_Flag=0#N_A=1AND#P ₁ =1:(Est_Num_Displ) IF#Rel_Flag=0#N_A=1AND#P ₁ =2:(Est_Num_Displ2).....IF#Rel_Flag=0#N_A=1AND #P ₁ =n:(Est_Num_Displn) IF#Rel_Flag=0#N_A=0AND#P ₁ =1:(Est_Anal_Displ1) IF#Rel_Flag=0#N_A=0AND#P ₁ =2:(Est_Anal_Displ2).....IF#Rel_Flag=0#N_A=0AND #P ₁ =n:(Est_Anal_Displn) IF#Rel_Flag=1#N_A=1AND#P ₁ =1:(Est_Num_Conf1) IF#Rel_Flag=1#N_A=1AND#P ₁ =2:(Est_Num_Conf2).....IF#Rel_Flag=1#N_A=1AND #P ₁ =n:(Est_Num_Confn) IF#Rel_Flag=1#N_A=0AND#P ₁ =1:(Est_Anal_Conf1) IF#Rel_Flag=1#N_A=0AND#P ₁ =2:(Est_Anal_Conf2).....IF#Rel_Flag=1#N_A=0AND #P ₁ =n:(Est_Anal_Confn)	---	1
t_{fi}	IF#Rel_Flag=0#N_A=1AND#P ₁ =1:1-(Est_Num_Displ1) IF#Rel_Flag=0#N_A=1AND#P ₁ =2:1-(Est_Num_Displ2)IF#Rel_Flag=0#N_A=1AND #P ₁ =n:1(Est_Num_Displn) IF#Rel_Flag=0#N_A=0AND#P ₁ =1:1-(Est_Anal_Displ1) IF#Rel_Flag=0#N_A=0AND#P ₁ =2:1-(Est_Anal_Displ2)IF#Rel_Flag=0#N_A=0AND #P ₁ =n:1-(Est_Anal_Displn) IF#Rel_Flag=1#N_A=1AND#P ₁ =1:1-(Est_Num_Conf1) IF#Rel_Flag=1#N_A=1AND#P ₁ =2:1-(Est_Num_Conf2).....IF#Rel_Flag=1#N_A=1AND #P ₁ =n:1-(Est_Num_Confn) IF#Rel_Flag=1#N_A=0AND#P ₁ =1:1-(Est_Anal_Conf1) IF#Rel_Flag=1#N_A=0AND#P ₁ =2:1-(Est_Anal_Conf2).....IF#Rel_Flag=1#N_A=0AND #P ₁ =n:1-(Est_Anal_Confn)	---	1
t_{fsi}	IF #Rel_Flag0:(Est_Indisponibilidade_Falha_Segura) IF#P ₁ =1AND#N_A1=1:(Est_FSeg1) IF#P ₁ =2AND#N_A1=1:(Est_FSeg2).....IF#P ₁ =nAND#N_A1=1:(Est_FSegn) IF#N_A1=0AND(#DF1=0OR#NMR1=1):1 IF#N_A1=0AND(#DF=1AND#NMR1=0):(C)ELSE1	---	1
t_{fui}	IF #Rel_Flag0:(Est_Indisponibilidade_Falha_Insegura) IF#P ₁ =1AND#N_A1=1:(Est_FISeg1) IF#P ₁ =2AND#N_A1=1:(Est_FISeg2).....IF#P ₁ =nAND#N_A1=1:(Est_FISegn) IF#N_A1=0:(1-C)ELSE0	---	1
t_t	#P1	#P ₁ =0 AND #P ₁ >0	2
t_{Udi}	1	---	1
t_{Fdi}	1	---	1
t_{Sdi}	1	---	1

As expressões de multiplicidade dos arcos de m_1 a m_9 da Tabela 5.22 indicam todas as combinações possíveis de um conjunto de blocos em série para liberação de um resultado sucesso, falha segura ou falha insegura. Para que um conjunto n_1 de blocos em série libere um resultado válido de confiabilidade ou disponibilidade, é necessário que todos os blocos de entrada do bloco de decisão estejam operacionais. Caso apenas um deles esteja em falha, a saída será inválida. Neste caso, o arco m_7 habilitará a transição t_{S_i} , se a quantidade de tokens no lugar S_i for n_1 (todos os blocos operacionais). Caso contrário o arco desabilitará a transição t_{S_i} . Para o caso de todos os blocos estarem operacionais, os lugares F_i e U_i não devem ter qualquer token. Logo $\#F_i$ e $\#U_i$ têm zero token como mostrado pelas expressões dos arcos m_8 e m_9 . As estatísticas de ordem, descritas no Capítulo 2, definem por meio de expressões analíticas configurações serial e paralela e m/n .

Na Tabela 5.23 são descritos os parâmetros de configuração do modelo de blocos seriais múltiplos, enquanto na Tabela 5.24 são mostradas as expressões lógicas condicionais dos pesos das transições imediatas em conflito t_{F_i} e t_{S_i} , as quais são dependentes da marcação, para definição das avaliações a serem executadas. Pode-se considerar o mesmo fator de cobertura C para todos os blocos em série, porém conforme está definido na expressão das transições de falha segura, $t_{F_{S_i}}$, e de falha insegura, $t_{F_{U_i}}$ da Tabela 5.24, cada bloco pode ter um valor específico para o fator de cobertura o qual é dependente do número de *tokens* em $\#P_1$. A quantidade de *tokens* em $\#P_1$ corresponde ao bloco que está sendo avaliado. A coluna P na Tabela 5.24 representa o nível de prioridade das diversas transições imediatas. As estimativas de confiabilidade, disponibilidade e segurança, além das estimativas de falhas seguras e inseguras podem ser obtidas por meio das métricas definidas na Tabela 5.21.

Na Tabela 5.24 são mostradas as possíveis expressões numéricas ou analíticas associadas a cada um dos blocos presentes no modelo serial múltiplo. Como nas redes EDSPN não há distinção da cor do *token*, a única maneira de se atribuir diferentes valores numéricos ou expressões analíticas a cada um dos blocos é por meio da marcação do lugar P_1 . Por exemplo, supondo que haja 3 blocos em série, na representação pelo modelo serial múltiplo com $n_1=3$, tem-se as seguintes situações: quando $\#P_1=3$, valores numéricos ou expressões analíticas são atribuídos ao terceiro bloco; quando $\#P_1=2$ valores numéricos ou expressões analíticas são atribuídos ao segundo bloco; e quando $\#P_1=1$ valores numéricos ou expressões analíticas são atribuídos ao primeiro bloco. A atribuição destes valores é essencial no cálculo das estimativas. O resultado das estimativas corresponderá ao produto dos pesos atribuídos a cada bloco, por meio das transições de sucesso (t_{S_i}), ou de falha (t_{F_i}).

5.2.6 Modelo MDP: Modelo Bloco Ativo

O modelo MDP de bloco ativo, semelhantemente ao modelo MDP dos blocos seriais múltiplos, são utilizados para minimizar o comprimento e a complexidade dos diagramas de sistemas dependáveis e parametrizados, pela transformação de um conjunto de blocos em paralelo em um único bloco composto. O modelo MDP de bloco ativo, também ser definido como modelo MDP de blocos paralelos múltiplos, representa blocos que atuam paralelamente, ou blocos que atuam concorrentemente por meio das configurações NMR e m/n . Cada bloco NMR ou m/n é composto por várias réplicas atuando concorrentemente e regras lógicas que definem a validade da saída do bloco de decisão, cujo mecanismo de votação é considerado perfeito. No caso da configuração NMR o número de réplicas é um número natural e ímpar, e no caso m/n o número de réplicas é um número natural qualquer. A regra de decisão para o bloco m/n utiliza um parâmetro q o qual indica a quantidade mínima de réplicas a partir da qual o bloco ativo torna-se falho. Ou seja, caso o bloco m/n tenha no mínimo $n-q$ réplicas confiáveis ele será confiável.

O modelo de bloco ativo é composto por modelos de blocos em paralelo concatenados a um bloco de decisão, conforme mostrado na Figura 5.13. A multiplicidade dos arcos, mostrada nas Tabelas 5.17, 5.18 e 5.19, determina as regras de decisão para n blocos concorrentes cujos parâmetros associados as taxas de falha e de reparo podem assumir quaisquer valores, podendo excepcionalmente assumir valores iguais. Regras de configuração paralela para um bloco ativo podem ser definidas por meio de atribuição do valor 0 (zero) ao *flag* #Ser_Par, ou um valor 1 para os *flags* #NMR ou #M_N nas tabelas do bloco de decisão. A composição dos *flags* na determinação das estimativas de dependabilidade pode ser obtida da Tabela 5.20, conforme definida anteriormente, para os blocos de decisão.

O número de blocos em paralelo é representado pelo número de *tokens* n , presentes em P_1 , após disparo do arco m_{10} , cuja expressão de multiplicidade apresenta um valor igual a n . Desse modo reduz-se o número de lugares, transições e arcos correspondentes aos n blocos em paralelo, associando-se a cada bloco um *token* [3] e utilizando-se as mesmas camadas de modelo e de configuração comuns a todos. Conforme já afirmado anteriormente, ao invés de criarem-se n modelos particulares para n blocos concorrentes, pode-se modelar o conjunto por um único modelo com n *tokens* no lugar inicial, representando os n blocos.

Considerações Finais

A combinação dos modelos MDP, correspondentes aos blocos básicos ou tolerantes a falhas do diagrama EDBD, permite a elaboração de diagramas dependáveis dos mais diversos sistemas e das mais variadas complexidades, como será mostrado nos estudos de caso, onde serão abordados um sistema de aviação, um sistema de disparo do motor de um foguete lançador de mísseis e um sistema de telecomunicações de suporte a um sistema elétrico. Os modelos MDP permitem que sejam feitas análises de dependabilidade dos diagramas de blocos tolerantes a falhas levando-se em conta taxas de falha e reparo idênticas para os blocos, como nos modelos EDSPN do nível hierárquico 4, ou taxas variadas. Esta possibilidade de análise de diversos blocos com taxas de falha e reparo variáveis, torna o modelo MDP flexível e interessante. No modelo MDP tanto a análise transiente, correspondente as estimativas de confiabilidade, quanto as estimativas de estado permanente, correspondentes a análise de disponibilidade são produzidas em função das estimativas analíticas e/ou numéricas obtidas dos modelos EDSPN do nível hierárquico 4, os quais são analisados de um modo isolado e independente. Com o modelo MDP podem-se estimar os requisitos do sistema como um todo, a partir das estimativas dos blocos individuais e da análise na forma produto, a qual é expressa como um produto de fatores descrevendo o estado de cada bloco, o que reduz a possibilidade de explosão de estados e de *stiffness*.

Capítulo 6

Refinamento dos modelos EDSPN

Introdução

A organização dos sistemas de uma forma modular, e a sua representação inicial por meio de diagramas de blocos, faz com que um maior detalhamento destes sistemas corresponda ao refinamento dos blocos que os compõem, e estes por sua vez ao refinamento dos seus componentes e ao modo como estes interagem. Refinamentos do processo de modelagem se tornam necessários a medida que a complexidade dos sistemas vão sendo melhor compreendidas. Quanto maior o nível de detalhamento dos sistemas, ou dos blocos que os representam, mais representativos serão os modelos EDSPN correspondentes a estes blocos, e mais próximos da realidade serão os resultados das avaliações de dependabilidade geradas a partir dos modelos MDP. Um maior detalhamento dos modelos EDSPN e MDP conduz a uma maior precisão dos resultados obtidos permitindo ao modelador tomar decisões de um modo mais coerente e preciso. Os refinamentos podem estar relacionados aos componentes de cada bloco, a estados, eventos e conexões. O ajuste fino das características associadas a estados, eventos, conexões, e tempos de vida e de reparo de cada bloco, estão relacionados nos modelos EDSPN, ao refinamento de lugares, transições, multiplicidade condicional de arcos e multiplicidade condicional dos tempos de disparo das transições temporizadas de falha e de reparo, respectivamente.

6.1 Modelo EDSPN: Bloco Básico sem Replicação - Refinamento dos Eventos de Falha e de Reparo (Distribuições Não-Markovianas)

No modelo EDSPN correspondente ao bloco básico, eventos de falha e reparo apresentam distribuições exponenciais. De modo a possibilitar um maior refinamento dos processos de falha e de reparo, pode-se considerar a divisão desses processos em diversos estágios, representados por seqüências de fases. O refinamento pode ser relativo apenas ao evento de reparo, ao evento de falha, ou a ambos os eventos. No exemplo mostrado na Figura 6.1, foram considerados os refinamentos dos eventos relativos a falha e ao reparo através de duas seqüências de fases. No caso do processo de reparo a primeira fase corresponde as etapas de detecção e localização da falha do bloco, enquanto a segunda fase corresponde ao processo de reparo propriamente dito. Observa-se na Figura 6.1 que os tempos de reparo do bloco básico sem replicação, nas duas fases, são exponencialmente distribuídos com médias $1/\mu_1$ e $1/\mu_2$. Por outro lado, no processo de falha, a primeira fase pode representar o tempo de vida do bloco sob inspeção de modo a permitir uma manutenção preventiva, enquanto a segunda fase do tempo de vida do bloco pode indicar a necessidade de uma manutenção corretiva. No caso do processo de falha,

os tempos de vida do bloco básico são exponencialmente distribuídos com taxas λ_1 e λ_2 . O modelo que representa esta situação é descrito na Figura 6.1.

As distribuições dos tempos de falha e de reparo do modelo da Figura 6.1 correspondem a uma distribuição hipoexponencial formado por dois estágios. A distribuição hipoexponencial apesar de ser um processo semi-Markoviano é uma instância da distribuição de estágios de Cox e, portanto pode ser transformada em uma distribuição CTMC homogênea [140]. Por meio de um diagrama de estados, o qual é composto pelos estados de funcionamento confiável do bloco, correspondentes as fases 1 e 2, pelos estados de detecção e recuperação da falha, e pelas equações de balanceamento de fluxo, chega-se as expressões de probabilidade de estado permanente, as quais são válidas para as análises de disponibilidade do bloco básico. Embora o modelo apresentado na Figura 6.1 seja válido para duas fases apenas, um número maior de fases é possível, conforme pode ser observado nas expressões da Tabela 6.1. As expressões descritas na Tabela 6.1 são válidas para l fases.

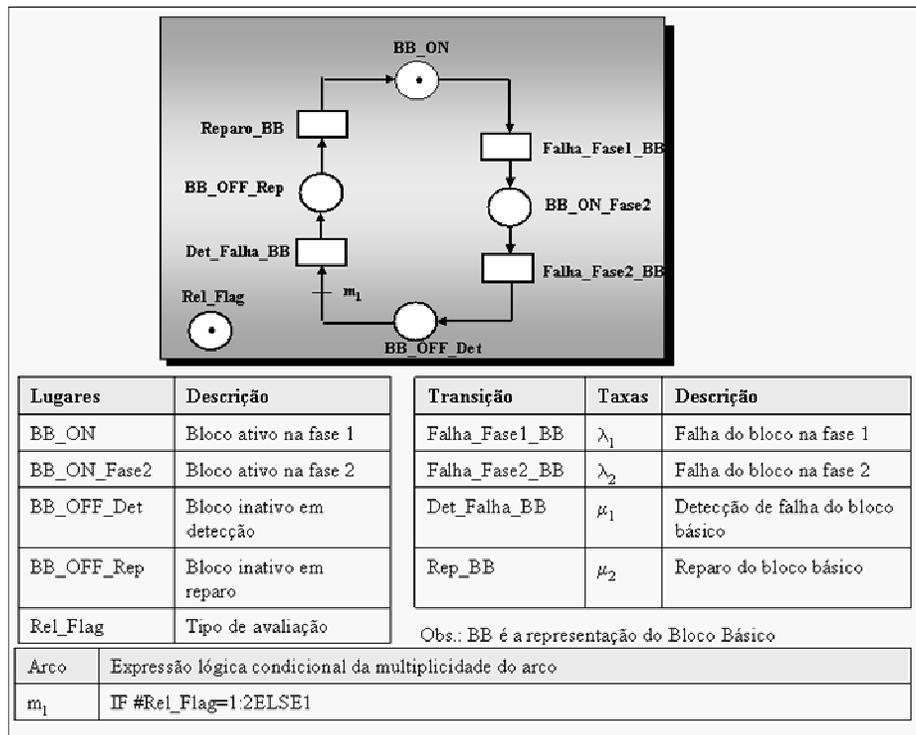


Figura 6.1 Modelo EDSPN de bloco básico com refinamento de evento

Observam-se nas expressões da Tabela 6.1, que a expressão de confiabilidade é representada por uma distribuição não-Markoviana do tipo hipoexponencial, composta por distribuições exponenciais com taxas $\lambda_1, \lambda_2, \dots, \lambda_l$. O processo de reparo constituído por l fases exponenciais, também tem o seu tempo de reparo representado por uma distribuição hipoexponencial. Portanto, a expressão de disponibilidade do bloco com refinamento dos eventos de falha e reparo apresenta uma relação entre taxas de falha e reparo equivalentes, relativas às distribuições hipoexponenciais, conforme descrito na Tabela 6.1.

Tabela 6.1 Expressões analíticas e métricas de dependabilidade para o bloco básico sem replicação e execução de falha e reparo em l fases.

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \sum_{\substack{j=1 \\ j \neq i}}^n a_i \exp^{-\lambda_i T}$, onde $a_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{\lambda_j}{\lambda_j - \lambda_i}$	$P\{\#BB_ON>0\}$
Inconfiabilidade	$Q(t) = \left(1 - \sum_{\substack{j=1 \\ j \neq i}}^n a_i \exp^{-\lambda_i T} \right)$, onde $a_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{\lambda_j}{\lambda_j - \lambda_i}$	$P\{\#BB_ON=0\}$
Disponibilidade	$A = \frac{1}{1 + \frac{\lambda_{equiv.}}{\mu_{equiv.}}}$ onde $\frac{1}{\mu_{equiv.}} = \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} + \dots + \frac{1}{\mu_l} \right) e$ $\lambda_{equiv.} = \frac{1}{(\lambda_1^{-1} + \lambda_2^{-1} + \dots + \lambda_l^{-1})}$	$P\{\#BB_ON>0\}$
Indisponibilidade	$U = 1 - A = 1 - \frac{1}{1 + \frac{\lambda_{equiv.}}{\mu_{equiv.}}}$	$P\{\#BB_ON=0\}$

Caso haja apenas uma única fase no processo de reparo e de falha, a expressão de confiabilidade se reduz a uma distribuição exponencial e os parâmetros da expressão de disponibilidade se reduzem a $\lambda_{equiv.} = \lambda$ e $\mu_{equiv.} = \mu$, onde λ e μ são as taxas de falha e de reparo das distribuições exponenciais de falha e de reparo do bloco básico.

6.2 Modelo EDSPN: Bloco Básico sem Replicação - Refinamento do Estado de Falha

No modelo de bloco básico sem replicação da Figura 5.1, o processo de detecção da falha é considerado perfeito. Acontece que nos sistemas reais nem sempre é possível detectar-se a presença de falhas quando de suas ocorrências. Esta probabilidade de detecção da falha quando de sua ocorrência é denominado fator de cobertura e neste modelo é representado pela letra C (*Coverage factor*). Com base nessa informação, pode-se inferir que falhas podem ou não ser detectadas, e este conhecimento conduz a um maior refinamento da natureza do evento de falha.

Considerando-se que a cobertura de falha seja imperfeita ($C < 1$), ou ainda que o processo de detecção de falha possa ou não ocorrer com probabilidades distintas, tem-se

que o tempo de reparo apresenta uma distribuição hiperexponencial com duas fases concorrentes e distintas. Numa das fases, com probabilidade α_1 , há a detecção da falha seguida pela realização de reparo; numa outra fase, com probabilidade α_2 , não há a detecção da falha ou esta é percebida após um tempo bastante longo quando de sua ocorrência. Uma vez percebida a falha inicia-se o processo de reparação.

O tempo despendido até a detecção da falha somado ao tempo de reparo propriamente dito é denominado tempo de percepção, e a taxa correspondente é denominada taxa de percepção. A Figura 6.2 descreve a possibilidade de refinamento de estado no modelo EDSPN de bloco básico. A cobertura de falha não perfeita do modelo da Figura 6.2 permite a obtenção de métricas de confiabilidade, disponibilidade e segurança.

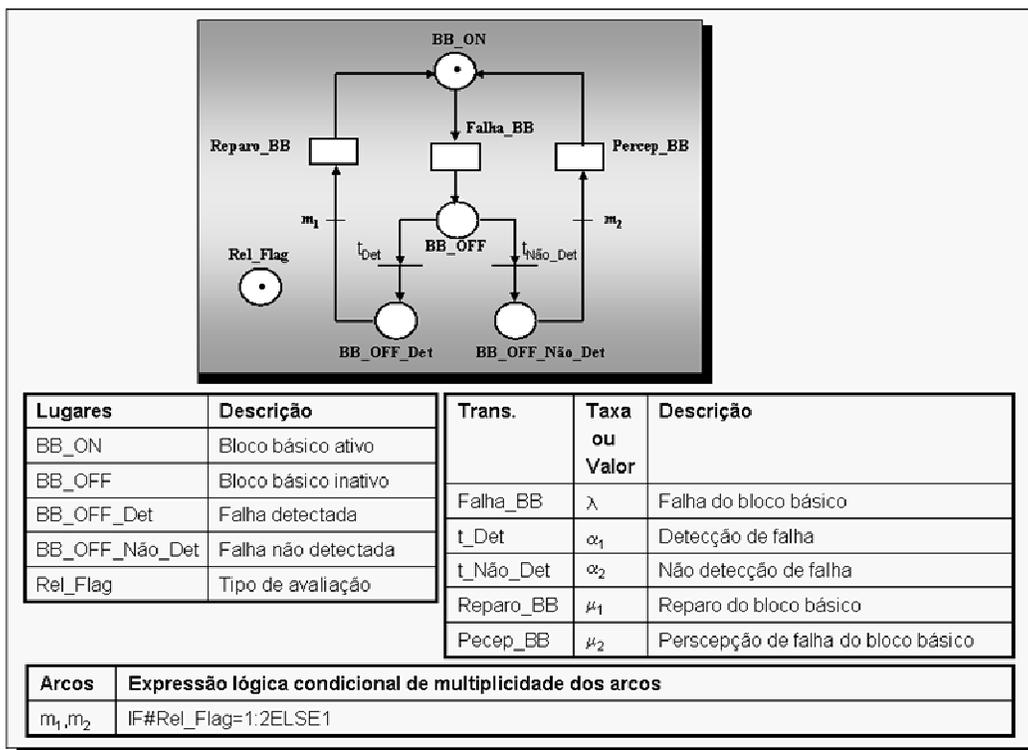


Figura 6.2 Modelo EDSPN de bloco básico sem replicação com refinamento de lugar

Através de um diagrama de estados da CTMC correspondente, composto pelos estados de funcionamento confiável, por estados de falha e pelas equações de balanceamento de fluxo, chega-se as expressões de probabilidade de estado permanente, as quais são válidas para as análises de disponibilidade do bloco básico.

Tabela 6.2 Expressões analíticas e métricas de dependabilidade para o bloco básico sem replicação e com cobertura de falhas.

Atributo	Expressões Analíticas	Métricas
Confiabilidade	$R(t) = \exp^{-\lambda t}$	$P\{\#BB_ON>0\}$
Inconfiabilidade	$Q(t) = (1 - \exp^{-\lambda t})$	$P\{\#BB_ON=0\}$
Falha Segura	$FS(t) = (C)\exp^{-\lambda t}$	$P\{\#BB_OFF_Det=1\}$
Falha Insegura	$FU(t) = (1 - C)\exp^{-\lambda t}$	$P\{\#BB_OFF-Not_Det\}=1$
Segurança	$S(t) = \exp^{-\lambda t} + C \exp^{-\lambda t}$	$P\{\#BB_ON+\#BB_OFF_Det=1\}$
Insegurança	$IS(t) = (1 - C)\exp^{-\lambda t}$	$P\{\#BB_OFF-Not_Det\}=1$
Disponibilidade	$A = \frac{1}{1 + \lambda \left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2} \right)}$	$P\{\#BB_ON>0\}$
Indisponibilidade	$U = 1 - \frac{1}{1 + \lambda \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} \right)}$	$P\{\#BB_ON=0\}$

6.3 Modelo EDSPN: Bloco Composto com Refinamento das Taxas de disparo das Transições Temporizadas de Reparo

Considerando-se a existência de blocos compostos, contendo até n blocos básicos operando em paralelo com uma taxa de falha comum e constante λ , podem-se arbitrar diferentes soluções de reparo e, por conseguinte, obter-se diferentes valores de disponibilidade. Neste modelo admite-se que haja variabilidade apenas das condições de reparo, uma vez que a condição de falha é referente a operação concorrente dos blocos. A mudança das condições de reparo é possível, por meio das alterações das expressões condicionais das taxas de disparo das transições temporizadas de reparo dependentes das marcações. Com o refinamento no dimensionamento da multiplicidade das taxas de reparo, diferentes soluções de reparo podem ser obtidas referentes às equipes de manutenção:

- **reparadores individuais** – os blocos em falha são reparados individualmente por equipes de manutenção distintas, cada qual, porém com a mesma taxa de reparo μ , ou seja, com o mesmo MTTR. Esta solução se assemelha a semântica de temporização *infinite server*, definida nas redes de fila e importada pelas redes de Petri;
- **reparador único** – todos os blocos pertencentes ao bloco composto têm uma solução de reparo comum e taxa de reparo μ . Nesta solução há apenas uma equipe de manutenção e, portanto todos os blocos em falha devem permanecer em fila até serem reparados, um a um. Esta solução equivale a semântica de temporização *single server* das redes de fila e definida nas redes de Petri;

- **múltiplos reparadores** – nesta solução existe um conjunto de m reparadores comuns a todos os n blocos básicos que compõem o bloco composto. Esta solução equivale a semântica de temporização *multiple server* das redes de fila a qual é utilizada nas redes de Petri.

No modelo EDSPN da Figura 6.3, define-se o refinamento das taxas de reparo da transição temporizada reparo_BB, através de expressões lógicas condicionais. Na Figura 6.3, pode-se observar que o modelo é composto por n blocos básicos, representado pela marcação n no lugar BB_ON, e por alguns lugares isolados, denominados *flags*, os quais são usados para definição da solução de reparo. Além dos *flags* na Figura 6.3, devem ser definidos os parâmetros MTTF e MTTR, correspondentes aos tempos médios para falha e reparo, além dos parâmetros n e m , correspondentes ao número de blocos concorrentes e ao número de reparadores.

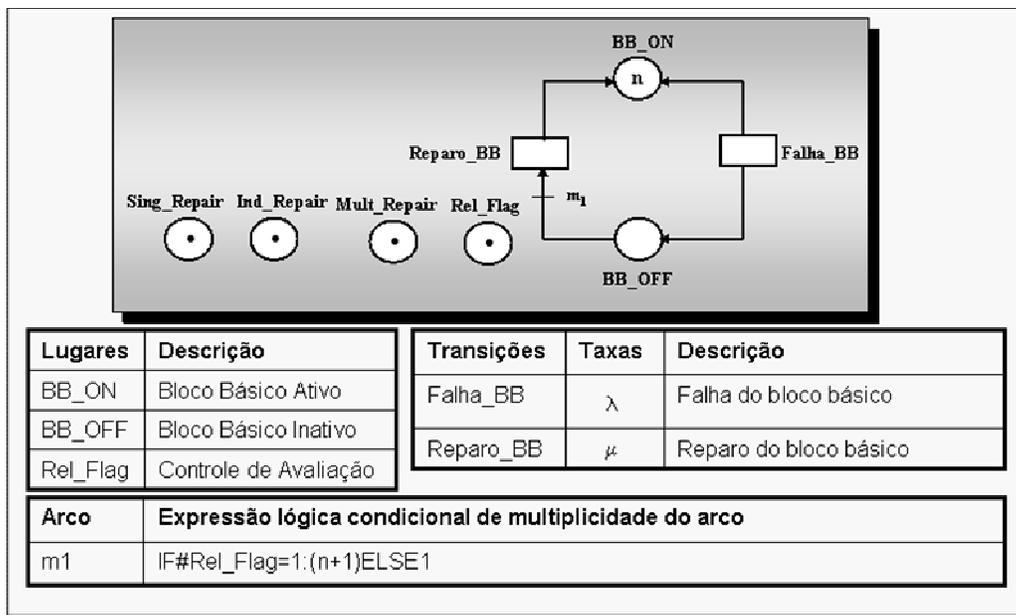


Figura 6.3 Modelo EDSPN para múltiplos blocos com diferentes soluções de reparo.

As expressões lógicas condicionais das múltiplas taxas ou tempos de reparo dependentes das marcações, dos *flags* e dos parâmetros estruturais são definidas na Tabela 6.3.

Tabela 6.3 Expressões lógicas condicionais dos tempos de recuperação

Transições	Expressões lógicas condicionais dos tempos de reparo	Semântica de tempo
Falha_BB	$1/\lambda$	<i>Infinite_Server</i>
Reparo_BB	IF((#Ind_Repair=1OR(#Mult_Repair=1AND#BB_OFF< m))AND#BB_OFF>0:1/(μ .(#BB_OFF))IF(#Mult_Repair=1AND#BB_OFF>0AND#BB_OFF>(m-1):1/(μ .m)IF#Sing_Repair=1AND#BB_OFF>0:1/ μ ELSE0	<i>Single-Server</i>

Obs.: m é o número de reparadores numa solução de múltiplos reparos

Tabela 6.4 Métricas de dependabilidade

Atributo	Métricas
Confiabilidade	$P\{\#BB_ON>0\}$
Inconfiabilidade	$P\{\#BB_ON=0\}$
Disponibilidade	$P\{\#BB_ON>0\}$
Indisponibilidade	$P\{\#BB_ON=0\}$

As expressões de confiabilidade ou de disponibilidade podem ser definidas de um modo analítico ou numérico. Por meio das métricas definidas na Tabela 6.4, valores numéricos de dependabilidade podem ser obtidos da Figura 6.4, e utilizados nos modelos MDP, levando-se em conta as diferentes soluções de reparo.

6.4 Modelo EDSPN: Bloco sem replicação - refinamento dos componentes de hardware e software

Levando-se em consideração as diversas possibilidades de replicação, descritas no capítulo 4, várias são as opções de hardware e software, ou simplesmente hw e sw, na formação de cada bloco funcional, alguns dos quais pode ser observado em [1][3]:

- componente de hw único sem componente de sw;
- múltiplos componentes de hw sem componente de sw;
- componente de sw único em componente de hw único;
- múltiplos componentes de sw em componente de hw único;
- componente de sw único em múltiplos componentes de hw;
- múltiplos componentes de sw em múltiplos componentes de hw.

As duas primeiras combinações, formadas apenas por componentes de hardware, foram descritas anteriormente no Capítulo 5..

As configurações de bloco único, constituídas por diferentes combinações de hw e sw, podem ainda levar em conta a possibilidade ou não de detecção de falhas, através dos fatores de cobertura, descritos anteriormente no processo de refinamento de estado (lugar).

6.4.1 Modelo Bloco Básico: Configuração Sw único em Hw único

Neste modelo considera-se o bloco como sendo formado por um componente único de sw sendo executado em um único componente de hw. Não está sendo levada em conta a possibilidade de cobertura de falhas, ou seja, considera-se que todas as falhas são passíveis de detecção, isto é, que a detecção de falhas é perfeita.

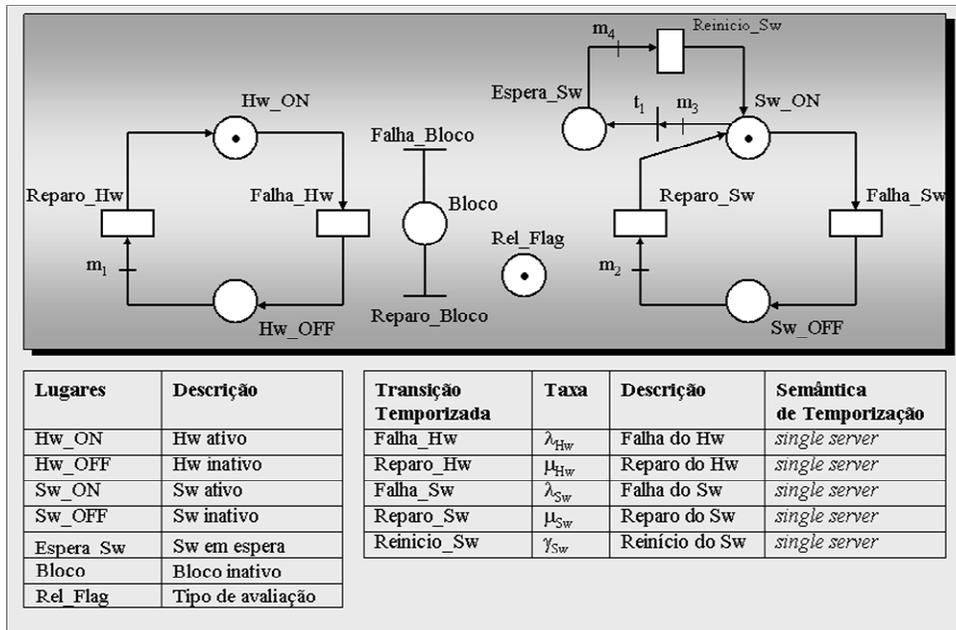


Figura 6.4 Modelo bloco único formado por hw único e sw único

A descrição dos lugares e transições temporizadas é expressa na Figura 6.4. As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade, podem ser observadas nas Tabelas 6.5, 6.6 e 6.7, respectivamente.

Tabela 6.5 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:2ELSE1
m ₃	IF#Hw_ON=0:1ELSE2
m ₄	IF#Rel_Flag=1:2ELSE1

Tabela 6.6 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
Falha_Bloco	(#Hw_ON=0OR#Sw_ON=0)AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>0)AND#Bloco=1	1

Tabela 6.7 Definição das métricas de dependabilidade

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$

No modelo de bloco básico as distribuições de falha, reparo e reinício são distribuições exponenciais, conforme mostrado na Figura 6.4. Neste modelo apenas as métricas de confiabilidade e disponibilidade podem ser avaliadas uma vez que o sistema não leva em consideração a cobertura de falhas, isto é, todas as ocorrências de falhas são admitidas serem detectadas. No próximo modelo a ser examinado, a cobertura de falhas é levada em consideração, o que possibilita a avaliação de outras métricas além da confiabilidade e disponibilidade.

6.4.2 Modelo Bloco Básico: Configuração Sw único, com Cobertura, em Hw único

Apesar do modelo de bloco básico com cobertura de falhas, possibilitar a detecção de falhas, tanto do hardware quanto do software, apenas as coberturas de falhas do software estão sendo consideradas, uma vez que a detecção de falhas em hardware seguem as mesmas considerações. Neste modelo, as distribuições de Falha_Hw, Falha_Sw, Reparo_Hw, Reparo_Sw e Reparo_Hw são distribuições exponenciais. Na Figura 6.5 o modelo é mostrado, conjuntamente com a descrição dos lugares e transições temporizadas que o compõe.

No modelo de bloco básico com cobertura de falhas, o percentual de detecção de falhas do componente de software é representado pelo fator de cobertura C , enquanto o percentual de não detecção de falhas é representado pelo complemento do fator de cobertura, ou seja, $1-C$. A possibilidade de detecção ou não de falhas no modelo é representado pelas transições imediatas em conflito t_{FS} e t_{FU} . Com a introdução do fator de cobertura de falhas, diversas outras métricas, além das métricas de confiabilidade e disponibilidade, relativas a falhas e segurança podem ser avaliadas, conforme pode ser visto na Tabela 6.10.

Tabela 6.8 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m_1	IF#Rel_Flag=1:2ELSE1
m_2	IF#Rel_Flag=1:2ELSE1
m_3	IF#Hw_ON=0:1ELSE2
m_4	IF#Rel_Flag=1:2ELSE1
m_5	IF#Rel_Flag=1:2ELSE1

Tabela 6.9 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t_1	1	1
Falha_Bloco	(#Hw_ON=0OR#Sw_ON=0)AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>0)AND#Bloco=1	1
t_{FS}	C	1
t_{FU}	$(1-C)$	1

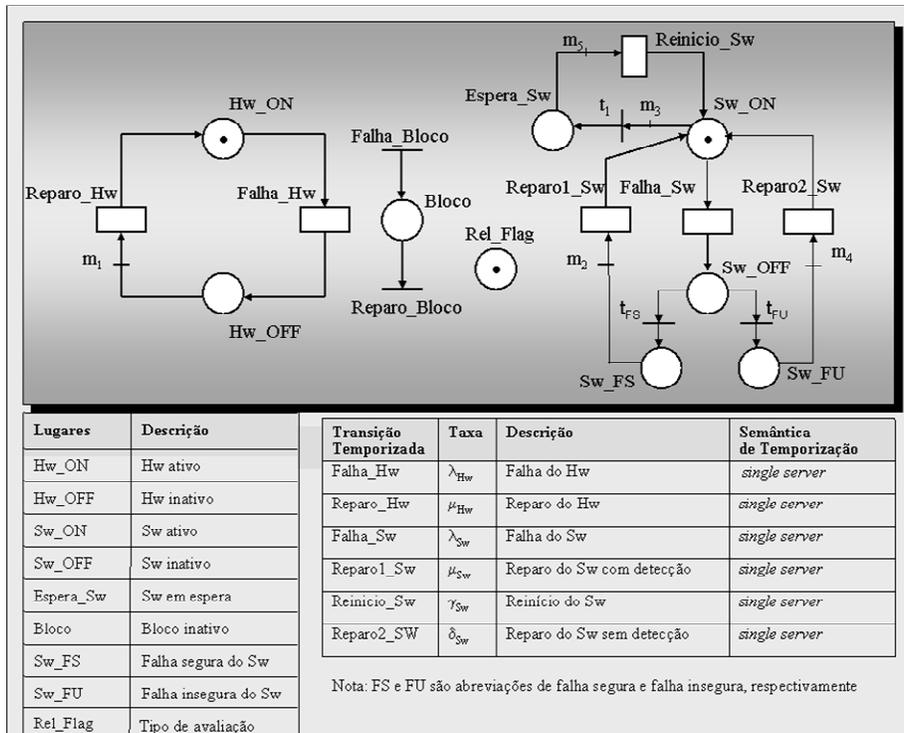


Figura 6.5 Modelo de bloco básico de hw e sw únicos com cobertura de falhas

Tabela 6.10 Definição das métricas com cobertura de falhas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$
Falha_Segura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } \#Sw_FS>0\}$
Falha_Segura_Hw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=0\}$
Falha_Insegura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } \#Sw_FU>0\}$
Segurança	$P\{(\#Bloco=1 \text{ AND } (\#Hw_ON=0 \text{ OR } (\#Hw_ON=1 \text{ AND } \#Sw_FS>0))) \text{ OR } \#Bloco=0\}$
Insegurança	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } \#Sw_FU>0\}$

6.4.3 Modelo Bloco Básico: Configuração Sw múltiplo, em Configuração NMR, e Hw único

O modelo de bloco básico com configuração NMR para o componente de software é apresentado na Figura 6.6. Este modelo é bastante semelhante ao modelo de bloco básico com componentes únicos de hardware e software. A diferença reside na quantidade de componentes de software no estado inicial, representado por $n1_Sw$ no lugar Sw_ON, e nas expressões de guarda e de multiplicidade de arcos. Aspectos de cobertura de falhas não estão sendo levados em consideração nesse modelo, ou seja, admite-se que todas as falhas sejam passíveis de detecção.

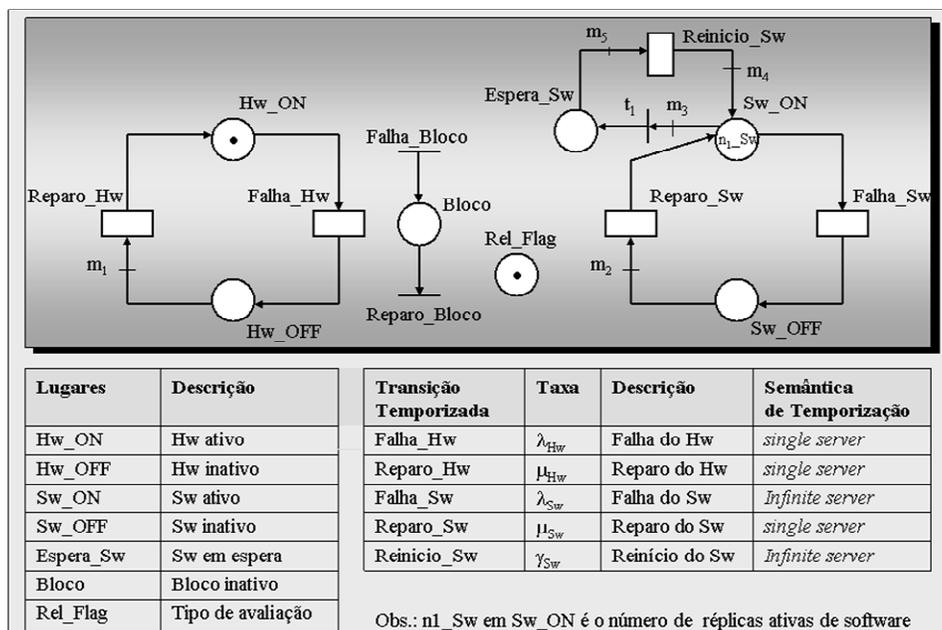


Figura 6.6 Modelo de bloco básico com Hw único e Sw em configuração NMR

A descrição dos lugares e transições temporizadas é expressa na Figura 6.6. As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade podem ser observadas nas Tabelas 6.11, 6.12 e 6.13, respectivamente. Semelhantemente ao modelo de bloco básico, as distribuições de falha, reparo e reinício são distribuições exponenciais, conforme mostrado na Figura 6.6. Nesse modelo avaliam-se apenas as métricas de confiabilidade e disponibilidade, uma vez que a detecção de falhas é considerada perfeita.

Tabela 6.11 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n _{1_Sw} +1)ELSE1
m ₃	IF#Hw_ON=0:1ELSE(n _{1_Sw} +1)
m ₄	IF#Rel_Flag=1:(n _{1_Sw} +1)ELSE1

Tabela 6.12 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
Falha_Bloco	(#Hw_ON=0OR(#Sw_ON<(n _{1_Sw} +1)/2)) AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>(n _{1_Sw} -1)/2)) AND#Bloco=1	1

Tabela 6.13 Definição das métricas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$

6.4.4 Modelo Bloco Básico: Configuração Sw múltiplo, em NMR com Cobertura, e Hw único

Neste tipo de modelo, mostrado na Figura 6.7, sobre o componente de hardware único são executadas concorrentemente várias réplicas de software dispostas numa configuração NMR. Neste caso, diferentemente do caso anterior, as réplicas de software estão sujeitas a cobertura de falhas, indicando que a detecção de falha da réplica de software poderá ou não ocorrer. Este modelo apesar de apresentar muitas semelhanças com o modelo de bloco básico, contém nas expressões condicionais de multiplicidade de arcos e nas funções de guarda suas principais diferenças.

As opções semânticas de temporização, esboçadas na Figura 6.7, definem o modo de disparo das diversas réplicas de Software. As transições de reparo com opção semântica *single server* indicam que a manutenção das réplicas sob reparo ocorrerão uma por vez. Por outro lado, opções semânticas do tipo *infinite server*, como no caso das transições de falha, permitem que mais de uma réplica de software possam falhar concorrentemente, uma vez que elas são executadas em paralelo.

As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade para o caso do bloco com configuração NMR e cobertura de falhas, podem ser observadas nas Tabelas 6.14, 6.15 e 6.16, respectivamente.

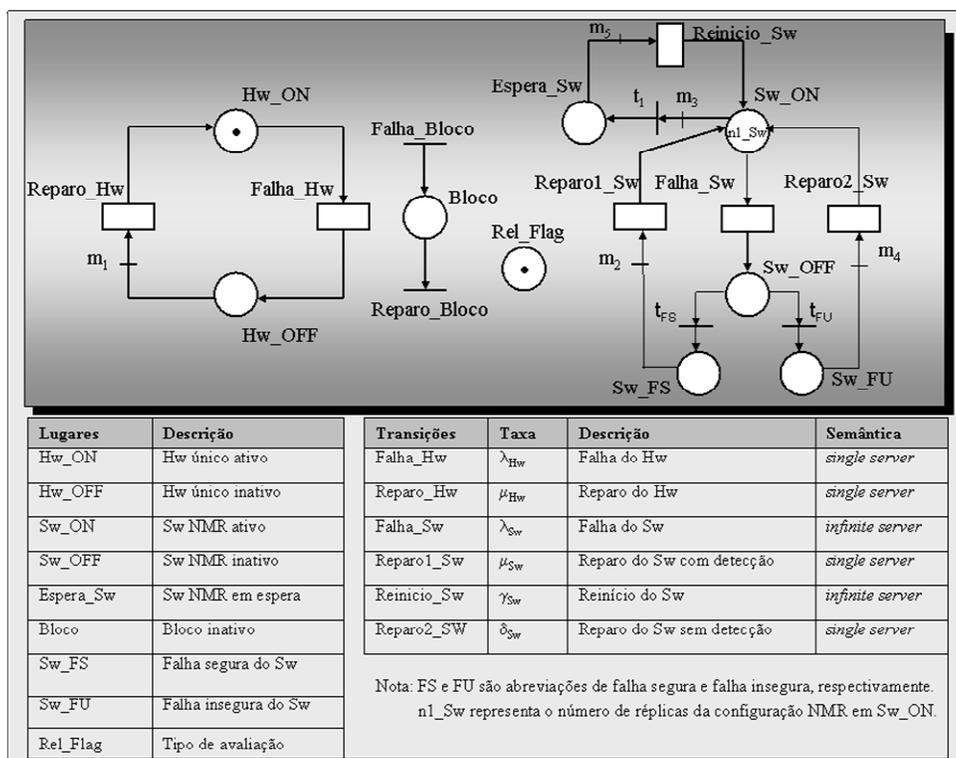


Figura 6.7 Modelo de bloco básico com Hw único e Sw NMR com cobertura

Tabela 6.14 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1
m ₃	IF#Hw_ON=0:1ELSE(n ₁ _Sw+1)
m ₄	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1
m ₅	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1

Tabela 6.15 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
tFS	C	1
tFU	(1-C)	1
Falha_Bloco	(#Hw_ON=0OR(#Sw_ON<(n ₁ _Sw+1)/2)) AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>(n ₁ _Sw-1)/2)) AND#Bloco=1	1

Tabela 6.16 Definição das métricas com cobertura de falhas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$
Falha_Segura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } (\#Sw_FS > \#Sw_FU) \text{ AND } (\#Sw_FS + \#Sw_FU > (n1_SW - 1)/2)\}$
Falha_Segura_Hw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=0\}$
Falha_Insegura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } (\#Sw_FU \geq \#Sw_FS) \text{ AND } (\#Sw_FS + \#Sw + FU > (n1 - Sw - 1)/2)\}$
Segurança	$P\{\#Bloco=1 \text{ AND } (\#Hw_ON=0 \text{ OR } (\#Hw_ON=1 \text{ AND } ((\#Sw_FS > \#Sw_FU) \text{ AND } (\#Sw_FS + \#Sw_FU > (n1_Sw - 1)/2)))) \text{ OR } \#Bloco=0\}$
Insegurança	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } (\#Sw_FU \geq \#Sw_FS) \text{ AND } (\#Sw_FS + \#Sw + FU > (n1 - Sw - 1)/2)\}$

Da mesma forma que o modelo de bloco básico, as distribuições de falha, reparo e reinício são distribuições exponenciais, conforme mostrado na Figura 6.7. Nesse modelo pode-se avaliar além das métricas de confiabilidade e disponibilidade, métricas relativas a detecção de falhas e métricas de segurança. Considera-se que todas as réplicas de software têm as mesmas taxas de falha, reparo e reinício, as quais são representadas por λ_{sw} , μ_{sw} , e γ_{sw} , respectivamente. Admite-se a taxa de percepção de falha β_{sw} neste modelo como sendo de um valor extremamente baixo, uma vez que os mecanismos de cobertura de falhas e autuação podem levar um tempo extremamente alto, ou mesmo podem não ser capazes de detectar algumas das falhas. O percentual de falhas detectadas, representada pelo fator de cobertura C , é utilizado como peso na transição imediata t_{FS} , enquanto o fator de não cobertura de falhas, representado pelo seu complemento $1-C$, é utilizado como peso na transição imediata t_{FU} .

6.4.5 Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (*ColdStandby*), e Hw único

No modelo de bloco básico composto por software com replicação passiva, do tipo *coldstandby* e hardware único, o componente de software é constituído por $n \geq 2$ réplicas sendo que apenas uma delas está ativa enquanto as demais estão inativas. A réplica de software ativa, executa os dados de entrada e está sujeita a falhas, enquanto as demais réplicas passivas, desenergizadas, não estão sujeitas a falhas. As réplicas inativas permanecem numa condição de espera até que uma delas seja solicitada a assumir a condição ativa em virtude de falha da réplica primária ativa. O modelo EDSPN do bloco com replicação *coldstandby* para o software é apresentado na Figura 6.8. Neste modelo não estão sendo levadas em conta a possibilidade de falhas não serem detectadas, ou seja, considera-se que todas as falhas são passíveis de detecção.

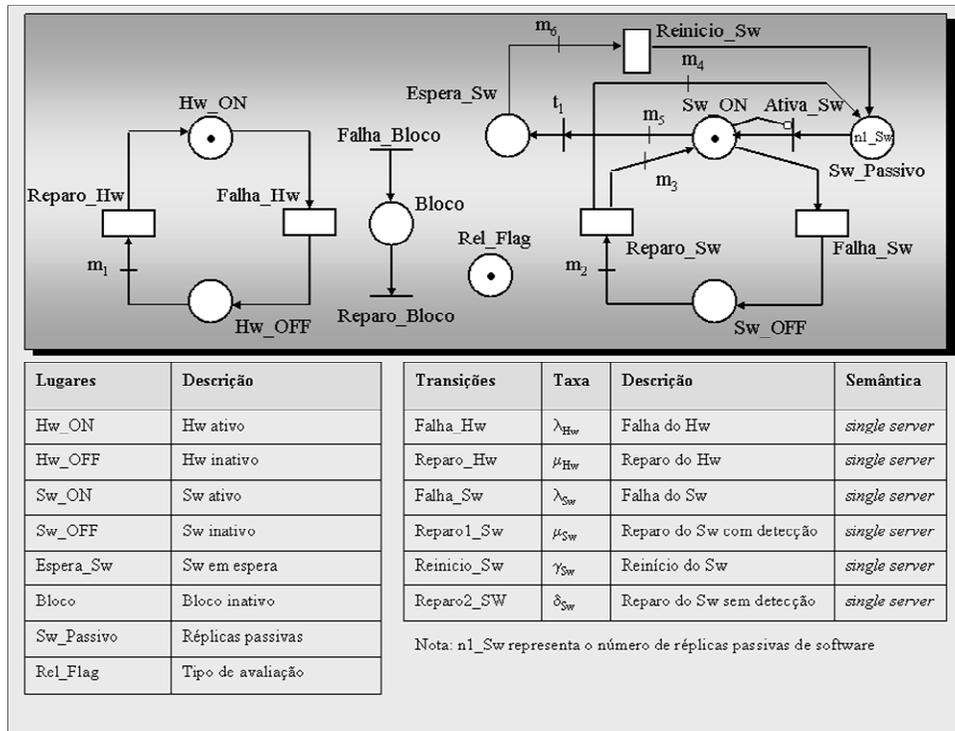


Figura 6.8 Modelo de bloco com Hw único e Sw com replicação *cold* e *warmstandby*

As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade para o caso do bloco com configuração do tipo *coldstandby* para o componente de software podem ser observadas nas Tabelas 6.17, 6.18 e 6.19, respectivamente.

Tabela 6.17 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m_1	$IF\#Rel_Flag=1:2ELSE1$
m_2	$IF\#Rel_Flag=1:(n1_Sw+2)ELSE1$
m_3	$IF\#Sw_ON=1:0ELSE1$
m_4	$IF\#Sw_ON=1:1ELSE0$
m_5	$IF\#Hw_ON=0:1ELSE(n1_Sw+2)$
m_6	$IF\#Rel_Flag=1:(n1_Sw+2)ELSE1$

Tabela 6.18 Pesos e funções de guarda das transições imediatas do modelo *coldstandby*

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
Ativa_Sw	1	1
Falha_Bloco	$((\#Hw_ON=0)OR(\#Sw_ON=0))AND\#Bloco=0$	1
Reparo_Bloco	$((\#Hw_ON>0)AND(\#Sw_ON>0))AND\#Bloco=1$	1

Tabela 6.19 Definição das métricas com cobertura de falhas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$

6.4.6 Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (*ColdStandby*) e Cobertura, e Hw único

Este modelo apresenta o mesmo tipo de replicação do componente de software do modelo anterior, porém desta vez com a possibilidade de falhas não serem detectadas. As políticas de reparo aplicam-se quando as falhas das réplicas são detectáveis. Caso alguma das réplicas venha a falhar e esta condição não seja detectada, uma outra réplica na condição passiva ficará impossibilitada de assumir a condição ativa pois os mecanismos de detecção e comutação não perceberão a falha. O tempo de retardo na detecção da falha somado ao tempo de reparo é denominado tempo de percepção.

O modelo de hardware único e réplicas de software na condição *coldstandby* é descrito na Figura 6.9. As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade para o caso do bloco com configuração do tipo *coldstandby* para o componente de software podem ser observadas nas Tabelas 6.20, 6.21 e 6.22, respectivamente.

Tabela 6.20 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n ₁ _Sw+2)ELSE1
m ₃	IF#Sw_ON=1:0ELSE1
m ₄	IF#Sw_ON=1:1ELSE0
m ₅	IF#Hw_ON=0AND#Rel_Flag=0:1ELSE(n ₁ _Sw+2)
m ₆	IF#Rel_Flag=1:(n ₁ _Sw+2)ELSE1
m ₇	IF#Sw_ON=1:0ELSE1
m ₈	IF#Sw_ON=1:1ELSE0
m ₉	IF#Hw_ON=0AND#Rel_Flag=1:(n ₁ _Sw+2)ELSE1
m ₁₀	IF#Hw_ON=0AND#Rel_Flag=1:(n ₁ _Sw+2)ELSE1
m ₁₁	IF#Rel_Flag=1:(n ₁ _Sw+2)ELSE1

Tabela 6.21 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
Ativa_Sw	1	2
tFS	C	3
tFU	(1-C)	3
Falha_Bloco	((#Hw_ON=0)OR(#Sw_ON=0))AND#Bloco=0	1
Reparo_Bloco	((#Hw_ON>0)AND(#Sw_ON>0))AND#Bloco=1	1

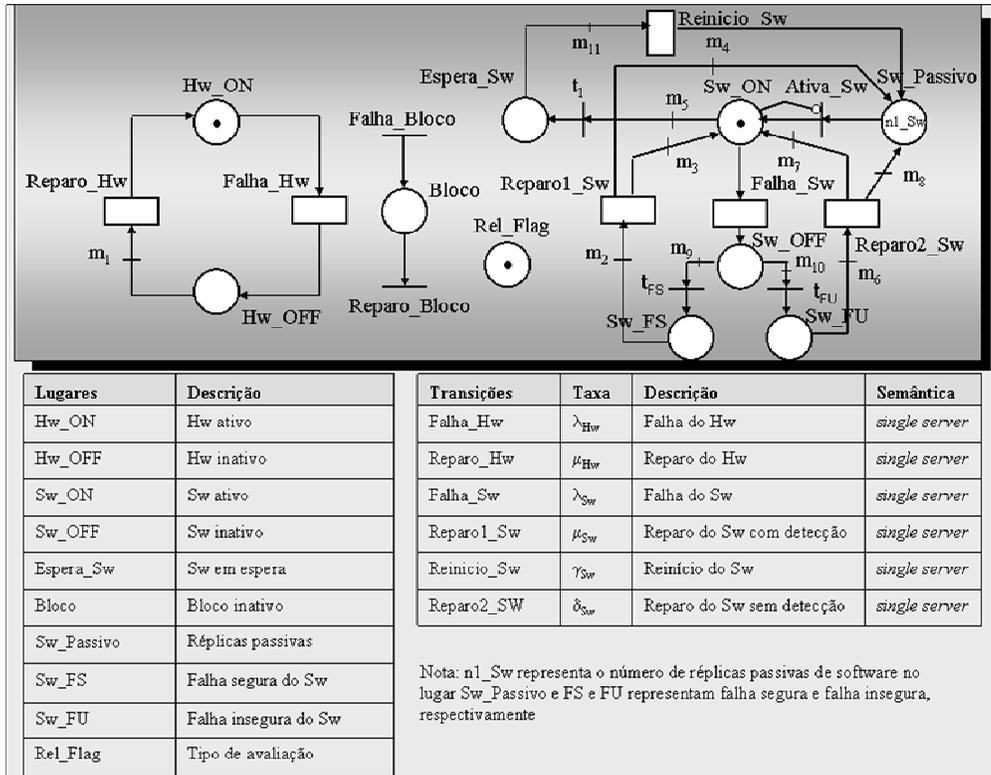


Figura 6.9 Bloco de Hw único e Sw com réplicas cold ou warmstandby com cobertura

Tabela 6.22 Definição das métricas com cobertura de falhas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$
Falha_Segura_Sw	$P\{\#Bloco=1AND(\#Hw_ON=0OR(\#Hw_ON=1AND\#Sw_FS=(n1_Sw+1)))\}$
Falha_Segura_Hw	$P\{\#Bloco=1AND\#Hw_ON=0\}$
Falha_Insegura_Sw	$P\{\#Sw_FU>0\}$
Segurança	$P\{(\#Bloco=1AND(\#Hw_ON=0OR(\#Hw_ON=1AND\#Sw_FS=(n1_Sw+1))))OR\#Bloco=0\}$
Insegurança	$P\{\#Sw_FU>0\}$

6.4.7 Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Passiva (*WarmStandby*), e Hw único

No modelo de bloco básico composto por replicação de software do tipo *warmstandby*, uma das réplicas de software está ativa enquanto as demais réplicas estão numa condição de espera passiva. As taxas de falha das réplicas ativa e passivas são diferentes, bem como o efeito de suas falhas. A taxa de falha da réplica de software ativa, a taxa de falha das réplicas de software passivas e as taxas de reparo das réplicas de software, são exponencialmente distribuídas com parâmetros λ_{sw} , v_w e μ_{sw} , respectivamente.

Neste modelo admite-se que todas as falhas sejam detectadas, e em caso de ocorrência de falha da réplica ativa, automaticamente uma das réplicas passivas armazenadas no lugar Sw_Passivo é comutada em seu lugar e a réplica que estava ativa vai para uma condição de falha no lugar Sw_OFF. As réplicas armazenadas no lugar Sw_OFF estão sujeitas a reparo.

O modelo apresentado na Figura 6.8 é válido para os blocos compostos com replicação de software do tipo *coldstandby* ou *warmstandby*, e hardware único. As Tabelas 6.17, 6.18 e 6.19 também são as mesmas para ambos os modelos. A diferença fundamental entre os modelos está relacionada ao tempo de retardo da transição temporizada *Falha_Sw*. No modelo *coldstandby* apenas a réplica ativa está sujeita a falha com taxa λ_{sw} , e portanto o tempo médio para falha MTTF é igual a $1/\lambda_{sw}$.

No modelo *warmstandby* a taxa de falha em qualquer instante é função da réplica ativa, com taxa de falha λ_{sw} , e das réplicas passivas, em atividade, com taxa de falha igual a v_{sw_i} onde $i=((n1_Sw), \dots, 2, 1)$ e $n1_Sw$ é o número inicial de réplicas passivas. Seguindo-se o modelo de estágios de COX na Figura 5.31, tem-se que o parâmetro MTTF no modelo *warmstandby* é dado pela expressão lógica condicional dependente da marcação do lugar Sw_OFF. Portanto a expressão do MTTF definida na transição *Falha_Sw* é dada por:

$$IF\#Sw_OFF = 0: \left(\frac{1}{(\lambda_{sw} + (v_{sw(n1_Sw)} + \dots + v_1))} \right) IF\#Sw_OFF = 1: \left(\frac{1}{(\lambda_{sw} + (v_{sw(n1_Sw-1)} + \dots + v_1))} \right)$$

$$IF\#Sw_OFF = 2: \left(\frac{1}{(\lambda_{sw} + (v_{sw(n1_Sw-2)} + \dots + v_1))} \right) \dots \dots IF\#Sw_OFF = n1_Sw: \left(\frac{1}{\lambda_{sw}} \right)$$

A medida que as réplicas falham aumenta a quantidade de réplicas em falha no lugar Sw_OFF e diminui a taxa de falha. No caso limite em que só há uma réplica ativa e nenhuma réplica passiva, a taxa de falha é dada por λ_{sw} e o tempo de retardo é especificado por $1/\lambda_{sw}$.

Caso as réplicas passivas apresentem a mesma taxa de falha v_{sw} a expressão do parâmetro MTTF na transição *Falha_Sw* é especificada por:

$$IF\#Sw_OFF = 0: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw)v_{Sw})} \right) IF\#Sw_OFF = 1: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw - 1)v_{Sw})} \right)$$

$$IF\#Sw_OFF = 2: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw - 2)v_{Sw})} \right) \dots\dots IF\#Sw_OFF = n1_Sw: \left(\frac{1}{\lambda_{Sw}} \right)$$

6.4.8 Modelo Bloco: Configuração Sw múltiplo, com Replicação Passiva (*WarmStandby*) e Cobertura, e Hw único

Neste modelo, como no modelo anterior, as réplicas ativa e passivas estão sujeitas a falhas. As falhas são detectadas com uma probabilidade C, e são passíveis de não detecção por um fator 1-C, onde C é o fator de cobertura. Este modelo apresenta o mesmo tipo de replicação do componente de software do modelo anterior, porém com a diferença que desta vez falhas podem não ser detectadas. As políticas de reparo são aplicáveis apenas quando as falhas das réplicas são detectáveis. Caso alguma das réplicas venha a falhar e esta condição não seja detectada, uma outra réplica na condição passiva ficará impossibilitada de assumir a condição ativa pois os mecanismos de detecção e comutação não perceberão a falha. O tempo de retardo na detecção da falha somado ao tempo de reparo é denominado tempo de percepção.

O modelo do bloco com hardware único e réplicas de software na configuração *warmstandby* é descrito pela Figura 6.9 e pelas Tabelas 6.20, 6.21 e 6.22. A única modificação está relacionada ao parâmetro MTTF correspondente a transição temporizada Falha_Sw. Neste caso, assim como no caso anterior sem cobertura, a expressão do parâmetro MTTF, considerando-se diferentes taxas de falha para as réplicas passivas é dada por:

$$IF\#Sw_OFF = 0: \left(\frac{1}{(\lambda_{Sw} + (v_{Sw(n1_Sw)} + \dots + v_1))} \right) IF\#Sw_OFF = 1: \left(\frac{1}{(\lambda_{Sw} + (v_{Sw(n1_Sw-1)} + \dots + v_1))} \right)$$

$$IF\#Sw_OFF = 2: \left(\frac{1}{(\lambda_{Sw} + (v_{Sw(n1_Sw-2)} + \dots + v_1))} \right) \dots\dots IF\#Sw_OFF = n1_Sw: \left(\frac{1}{\lambda_{Sw}} \right)$$

Caso as réplicas passivas apresentem a mesma taxa de falha v_{Sw} a expressão do parâmetro MTTF na transição Falha_Sw é especificada por:

$$IF\#Sw_OFF = 0: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw)v_{Sw})} \right) IF\#Sw_OFF = 1: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw - 1)v_{Sw})} \right)$$

$$IF\#Sw_OFF = 2: \left(\frac{1}{(\lambda_{Sw} + (n1_Sw - 2)v_{Sw})} \right) \dots\dots IF\#Sw_OFF = n1_Sw: \left(\frac{1}{\lambda_{Sw}} \right)$$

6.4.9 Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Semi-ativa (*HotStandby*), e Hw único

No modelo de bloco básico composto por replicação de software do tipo *hotstandby*, todas as réplicas de software estão concorrentemente ativas, porém só uma delas libera os resultados para a saída. As falhas das réplicas ativa e passivas são exponencialmente distribuídas com taxa de falha λ_{Sw} . O modelo com hardware único e réplicas de software na configuração *hotstandby* é mostrado na Figura 6.10.

O modelo de bloco básico com configuração *hotstandby* para o componente de software é apresentado na Figura 6.10. Este modelo é bastante semelhante ao modelo de bloco básico com componentes únicos de hardware e software. A diferença básica reside na quantidade de componentes de software no estado inicial, representado por $n1_Sw$ no lugar *Sw_ON*, e nas expressões de guarda e de multiplicidade de arcos. Aspectos de cobertura de falhas não são levados em consideração nesse modelo, ou seja, admite-se que todas as falhas são detectadas.

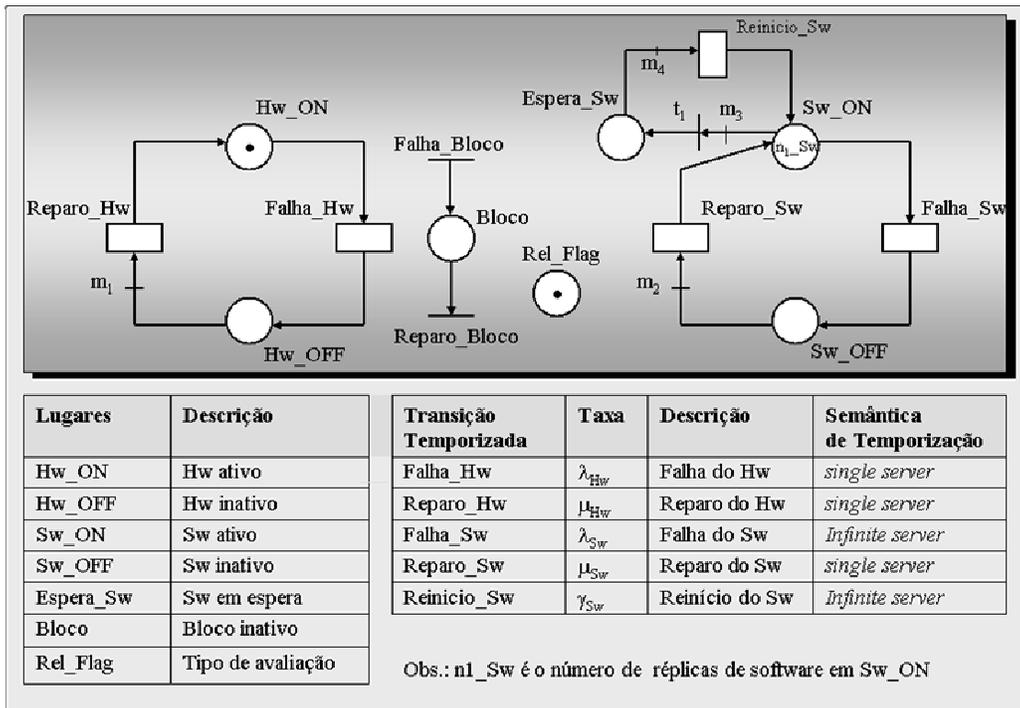


Figura 6.10 Modelo de bloco com Hw único e Sw com replicação *hotstandby*

A descrição dos lugares e transições temporizadas é expressa na Figura 6. 10. As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade podem ser observadas nas Tabelas 6.23, 6.24 e 6.25, respectivamente.

Tabela 6.23 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1
m ₃	IF#Hw_ON=0:1ELSE(n ₁ _Sw+1)
m ₄	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1

Tabela 6.24 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t1	1	1
Falha_Bloco	(#Hw_ON=0OR#Sw_ON=0)AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>0)AND#Bloco=1	1

Tabela 6.25 Definição das métricas

Métricas	Descrição
Confiabilidade	P{#Bloco=0}
Inconfiabilidade	P{#Bloco=1}
Disponibilidade	P{#Bloco=0}
Indisponibilidade	P{#Bloco=1}

O modelo EDSPN correspondente ao bloco de hardware único e réplicas de software na configuração *hotstandby*, ficará desabilitado quando o hardware falhar ou houver falha em todas as réplicas de software. O bloco voltará a condição ativa quando o hardware estiver ativo e pelo menos uma das réplicas de software também estiver ativa. As situações de falha e de reinício das réplicas de software ocorrem concorrentemente, conforme observado na Figura 6.10, e é representado pela semântica *infinite server*. Neste modelo os mecanismos de detecção de falhas são considerados perfeitos, isto é, as falhas são sempre detectadas.

6.4.10 Modelo Bloco Básico: Configuração Sw múltiplo, com Replicação Semi-ativa (*hotstandby*) com Cobertura, e Hw único

Neste tipo de modelo, mostrado na Figura 6.11, sobre o componente de hardware único são executadas concorrentemente várias réplicas de software dispostas numa configuração *hotstandby*. Neste caso, diferentemente do caso anterior, as réplicas de software estão sujeitas à cobertura de falhas, indicando que a detecção de falha da réplica de software poderá ou não ocorrer.

As opções semânticas de temporização, esboçadas na Figura 6.11, definem o modo de disparo das diversas réplicas de Software. As transições de reparo com opção semântica *single server*, indicam que a manutenção das réplicas sob reparo ocorrerão uma por vez. Por outro lado, opções semânticas do tipo *infinite server*, como no caso da transição Falha_Sw, permitem que mais de uma réplica de software possam falhar ao mesmo tempo uma vez que elas são executadas em paralelo.

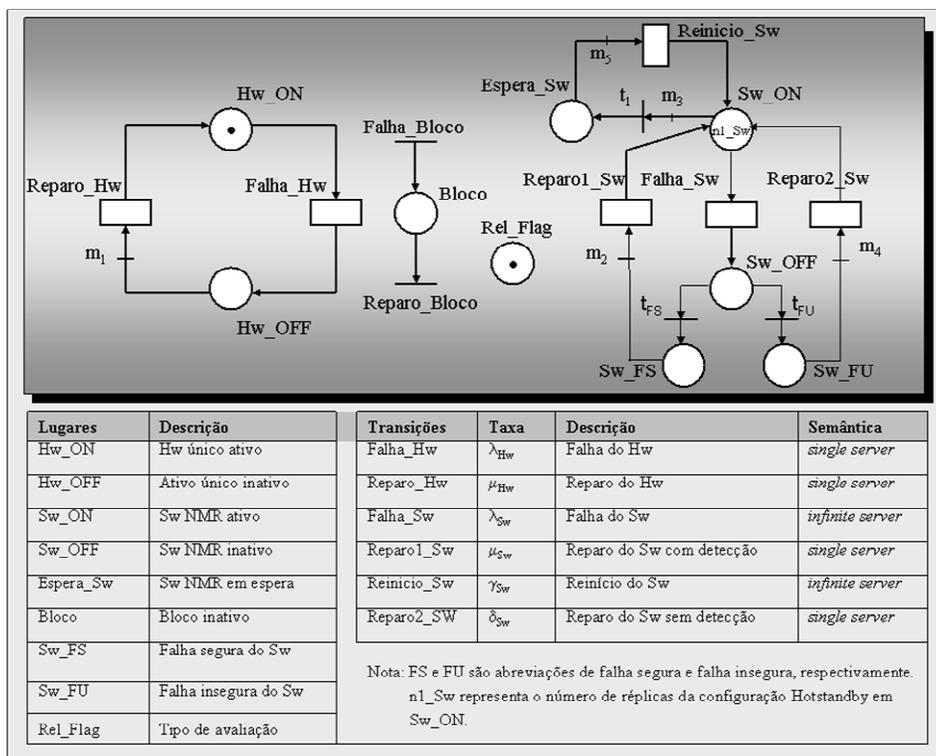


Figura 6.11 Modelo de bloco básico com Hw único e réplicas de Sw em hotstandby

As expressões condicionais das multiplicidades dos arcos dependentes das marcações, as expressões dos pesos e funções de guarda das transições imediatas e as definições das métricas de dependabilidade para o caso do bloco com configuração de software *hotstandby* e cobertura de falhas podem ser observadas nas Tabelas 6.26, 6.27 e 6.28, respectivamente.

Tabela 6.26 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1
m ₃	IF#Hw_ON=0:1ELSE(n ₁ _Sw+1)
m ₄	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1
m ₅	IF#Rel_Flag=1:(n ₁ _Sw+1)ELSE1

Tabela 6.27 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	Prioridade
t ₁	1	1
t _{FS}	C	1
t _{FU}	(1-C)	1
Falha_Bloco	(#Hw_ON=0OR#Sw_ON=0)AND#Bloco=0	1
Reparo_Bloco	(#Hw_ON>0AND#Sw_ON>0)AND#Bloco=1	1

Nota: C é a representação do fator de cobertura de falhas

Tabela 6.28 Definição das métricas com cobertura de falhas

Métricas	Descrição
Confiabilidade	$P\{\#Bloco=0\}$
Inconfiabilidade	$P\{\#Bloco=1\}$
Disponibilidade	$P\{\#Bloco=0\}$
Indisponibilidade	$P\{\#Bloco=1\}$
Falha_Segura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } \#Sw_FS=n1_Sw\}$
Falha_Segura_Hw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=0\}$
Falha_Insegura_Sw	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } (\#Sw_FU>0)\}$
Segurança	$P\{\#Bloco=1 \text{ AND } (\#Hw_ON=0 \text{ OR } (\#Hw_ON=1 \text{ AND } \#Sw_FS=n1_Sw)) \text{ OR } \#Bloco=0\}$
Insegurança	$P\{\#Bloco=1 \text{ AND } \#Hw_ON=1 \text{ AND } (\#Sw_FU>0)\}$

Do mesmo modo que o modelo de bloco básico, as distribuições de falha, reparo e reinício são distribuições exponenciais, conforme mostrado na Figura 6.11. Nesse modelo pode-se avaliar além das métricas de confiabilidade e disponibilidade, métricas relativas à detecção de falhas e métricas de segurança. As taxas de falha, reparo e reinício são representadas por λ_{Sw} , μ_{Sw} , e γ_{Sw} , respectivamente. Admite-se a taxa de percepção de falha β_{Sw} neste modelo como sendo de um valor extremamente baixo, uma vez que os mecanismos de cobertura de falhas e autuação podem levar um tempo extremamente alto, ou mesmo podem não ser capazes de detectar algumas das falhas. O percentual de falhas detectadas, representada pelo fator de cobertura C , é utilizado como peso na transição imediata t_{FS} , enquanto o fator de não cobertura de falhas, representado pelo seu complemento $(1-C)$, é utilizado como peso na transição imediata t_{FU} .

6.4.11 Modelo Bloco Básico: Configuração Sw múltiplo e Hw múltiplo

Várias são as combinações de múltiplos componentes de software e de hardware na composição de um dado bloco. Caso os múltiplos componentes apresentem comportamentos ou funcionalidades diversas, o modelo que os representa será composto por tantos modelos individuais quantos forem as funcionalidades. Caso alguns componentes apresentem o mesmo comportamento ou funcionalidade, eles serão agrupados em um único componente com tantos *tokens* quantos forem os componentes com o mesmo comportamento, permanecendo os demais componentes com diferentes comportamentos representados por modelos individuais [3].

A Figura 6.12 apresenta um bloco composto por dois componentes de hardware, cada qual com um componente de software associado. Caso cada componente de software apresente um comportamento distinto, considera-se a quantidade de *tokens* nos lugares $Sw1_ON$ e $Sw2_ON$ como sendo formada por apenas um único *token*. Logo, a quantidade de *tokens* nestes lugares, representada por $n1_Sw1$ e $n1_Sw2$ terá apenas um único *token*. Caso vários software apresentem o mesmo comportamento para um mesmo hardware, os valores de $n1_Sw1$ e $n1_Sw2$ na Figura 6.12 serão maiores do que 1 e a configuração desses componentes de software são representadas pelas funções de guarda nos blocos 1 e 2, respectivamente. Pode-se considerar, por exemplo, que os componentes

de software do bloco 1 estão em paralelo, enquanto os componentes de software do bloco 2 estão em série. Nas Tabelas 6.29 e 6.30 são especificados os lugares e as transições temporizadas do modelo de múltiplos componentes. A configuração dos componentes que formam o bloco como um todo, por hipótese, é definida como sendo paralela, representado pelas funções de guarda do bloco equivalente, conforme mostrado na Tabela 6.31. Ou seja, Bloco1 e Bloco2 estão configurados em paralelo na formação do Bloco_Equivalente.

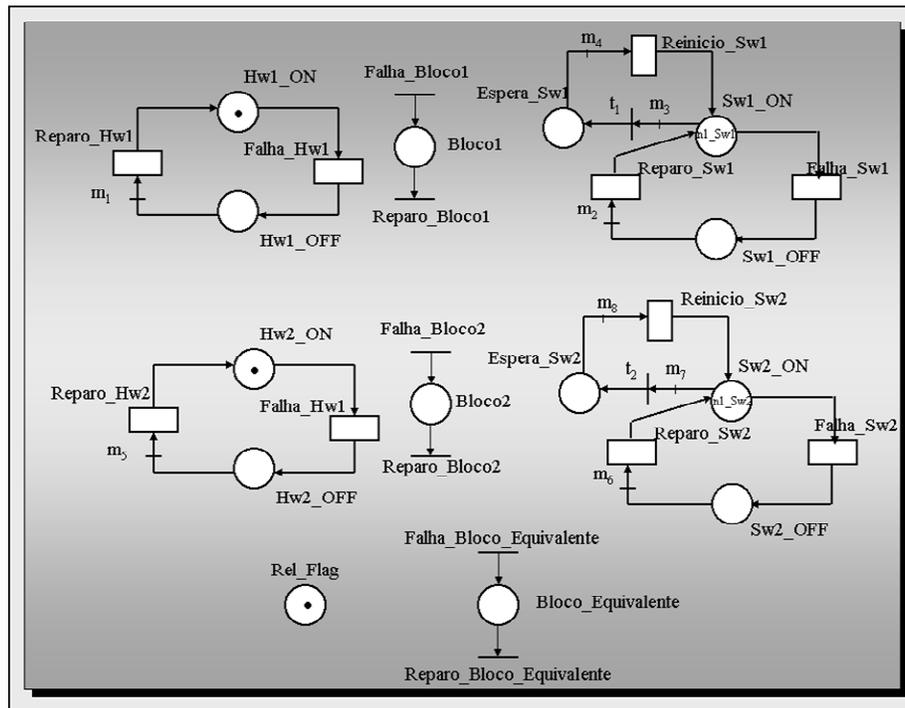


Figura 6.12 Modelo de Bloco básico com Hw e Sw múltiplos

Tabela 6.29 Lugares do modelo de blocos múltiplos

Lugares	Descrição
Hw1_ON	Componente Hw1 ativo
Hw1_OFF	Componente Hw1 inativo
Sw1_ON	Componente Sw1 ativo
Sw1_OFF	Componente Sw1 inativo
Espera_Sw1	Componente Sw1 em espera
Bloco1	Componente Bloco1 inativo
Hw2_ON	Componente Hw2 ativo
Hw2_OFF	Componente Hw2 inativo
Sw2_ON	Componente Sw2 ativo
Sw2_OFF	Componente Sw2 inativo
Espera_Sw2	Componente Sw2 em espera
Bloco2	Componente Bloco2 inativo
Bloco_Equiv.	Componente Bloco inativo
Rel_Flag	Condição de avaliação

Tabela 6.30 Transições temporizadas do modelo de blocos múltiplos

Transição	Taxa	Descrição	Semântica de Temporização
Falha_Hw1	λ_{Hw1}	Falha do Hw1	<i>Single Server</i>
Reparo_Hw1	μ_{Hw1}	Reparo do Hw1	<i>Single Server</i>
Falha_Sw1	λ_{Sw1}	Falha do Sw1	<i>Infinite Server</i>
Reparo_Sw1	μ_{Sw1}	Reparo do Sw1	<i>Single Server</i>
Reinício_Sw1	γ_{Sw1}	Reinício do Sw1	<i>Infinite Server</i>
Falha_Hw2	λ_{Hw2}	Falha do Hw2	<i>Single Server</i>
Reparo_Hw2	μ_{Hw2}	Reparo do Hw2	<i>Single Server</i>
Falha_Sw2	λ_{Sw2}	Falha do Sw2	<i>Single Server</i>
Reparo_Sw2	μ_{Sw2}	Reparo do Sw2	<i>Single Server</i>
Reinício_Sw2	γ_{Sw2}	Reinício do Sw2	<i>Single Server</i>

Tabela 6.31 Pesos e funções de guarda das transições imediatas

Transição	Expressão dos pesos ou funções de guarda	P
t1	1	1
Falha_Bloco1	$(\#Hw1_ON=0 \text{ OR } \#Sw1_ON=0) \text{ AND } \#Bloco1=0$	1
Reparo_Bloco1	$(\#Hw1_ON>0 \text{ AND } \#Sw1_ON>0) \text{ AND } \#Bloco1=1$	1
t2	1	1
Falha_Bloco2	$(\#Hw2_ON=0 \text{ OR } \#Sw2_ON<(n1_Sw2)) \text{ AND } \#Bloco2=0$	1
Reparo_Bloco2	$(\#Hw2_ON>0 \text{ AND } \#Sw2_ON=(n1_Sw2)) \text{ AND } \#Bloco2=1$	1
Falha_Bloco_ Equivalente	$(\#Bloco1>0 \text{ AND } \#Bloco2>0) \text{ AND } (\#Bloco_Equivalente=0)$	1
Reparo_Bloco_ Equivalente	$(\#Bloco1=0 \text{ OR } \#Bloco2=0) \text{ AND } (\#Bloco_Equivalente=1)$	1

Onde P é a abreviatura de Prioridade

As expressões condicionais das multiplicidades dos arcos dependentes das marcações e as definições das métricas de dependabilidade para o caso do bloco com múltiplos componentes podem ser observadas nas Tabelas 6.32 e 6.33, respectivamente.

Tabela 6.32 Multiplicidade dos arcos dependentes das marcações

Arcos	Expressões lógicas condicionais das multiplicidades dos arcos
m ₁	IF#Rel_Flag=1:2ELSE1
m ₂	IF#Rel_Flag=1:(n ₁ _Sw1+1)ELSE1
m ₃	IF#Hw_ON=0AND#Sw1_ON>0:(#Sw1_ON)ELSE(n ₁ _Sw1+1)
m ₄	IF#Rel_Flag=1:(n ₁ _Sw1+1)ELSE1
m ₅	IF#Rel_Flag=1:2ELSE1
m ₆	IF#Rel_Flag=1:(n ₁ _Sw2+1)ELSE1
m ₇	IF#Hw_ON=0:1ELSE(n ₁ _Sw2+1)
m ₈	IF#Rel_Flag=1:(n ₁ _Sw2+1)ELSE1

Tabela 6.33 Definição das métricas

Métricas	Descrição
Confiabilidade Bloco1	$P\{\#Bloco1=0\}$
Inconfiabilidade Bloco1	$P\{\#Bloco1=1\}$
Disponibilidade Bloco1	$P\{\#Bloco1=0\}$
Indisponibilidade Bloco1	$P\{\#Bloco1=1\}$
Confiabilidade Bloco2	$P\{\#Bloco2=0\}$
Inconfiabilidade Bloco2	$P\{\#Bloco2=1\}$
Disponibilidade Bloco2	$P\{\#Bloco2=0\}$
Indisponibilidade Bloco2	$P\{\#Bloco2=1\}$
Confiabilidade Bloco Equivalente	$P\{\#Bloco_Equivalente=0\}$
Inconfiabilidade Bloco Equivalente	$P\{\#Bloco_Equivalente=1\}$
Disponibilidade Bloco Equivalente	$P\{\#Bloco_Equivalente=0\}$
Indisponibilidade Bloco Equivalente	$P\{\#Bloco_Equivalente=1\}$

Considerações Finais

Neste capítulo foram descritas uma série de refinamentos: refinamentos associados a estados através, por exemplo, da troca de lugares que representam a detecção perfeita de falha, por lugares e transições que representam a detecção e a não detecção da falha, de acordo com o parâmetro de cobertura de falha; refinamentos associados a eventos pela troca das transições temporizadas associadas a falhas e reparo, por seqüências de transições exponenciais cujas fases descrevem cada etapa do processo de uma forma mais detalhada; refinamentos relativos a estratégias de manutenção através de diversas soluções de reparo por meio de expressões lógicas condicionais da multiplicidade dos tempos de reparo; refinamentos associados aos componentes de hardware e software que formam os blocos, onde foram descritas soluções conjuntas de hardware e software levando-se em conta as diversas formas de redundância e as possibilidades de cobertura de falhas. As diversas formas de refinamento aqui propostas, conjuntamente com os modelos básicos EDSPN e os modelos dependáveis e parametrizados (MDP), cobrem um grande número de sistemas reais.

Capítulo 7

Apresentação de Estudos de Caso

Introdução

Nos capítulos anteriores foram apresentadas diversas técnicas utilizadas nos projetos e análises de sistemas tolerantes a falhas. A avaliação e modelagem dos sistemas dependáveis seguem a metodologia esboçada no Capítulo 4, a qual faz uso dos modelos EDSPN e dos modelos MDP. Neste capítulo serão descritos alguns estudos de caso que permitirão validar a metodologia e os modelos considerados. Os dois primeiros estudos de caso estão relacionados a sistemas já existentes e em funcionamento, enquanto o terceiro estudo de caso diz respeito a um sistema na fase de projeto com mais de uma solução de arquitetura possível para a sua implementação. O quarto e último estudo de caso está relacionado à implementação de um sistema de ordenação tolerante a falhas que utiliza componentes de hardware e de software na sua estrutura.

7.1 Estudo de Caso I: Circuito de Disparo do Motor de um Foguete lançador de mísseis

O presente estudo de caso define estimativas de dependabilidade para um circuito eletrônico responsável pelo disparo do motor de um foguete lançador de mísseis. O circuito eletrônico de controle de partida do motor só deve ser acionado quando algumas condições forem satisfeitas, conforme descrição mais adiante. Contudo, este circuito pode ser detonado inadvertidamente, por acionamento prematuro, devido a falhas dos dispositivos eletrônicos que o compõem, interferências eletromagnéticas (EMI), especialmente devido a exposição a frequências de rádio (RF), ou por fatores externos tais como choque e temperaturas elevadas.

Para que se possa modelar a dependabilidade do sistema deve-se, inicialmente, desenvolver um diagrama de blocos, conforme mostrado na Figura 7.1, o qual define caminhos e indica dependências funcionais entre os elementos do sistema, necessários ao disparo, correto ou involuntário, do motor.

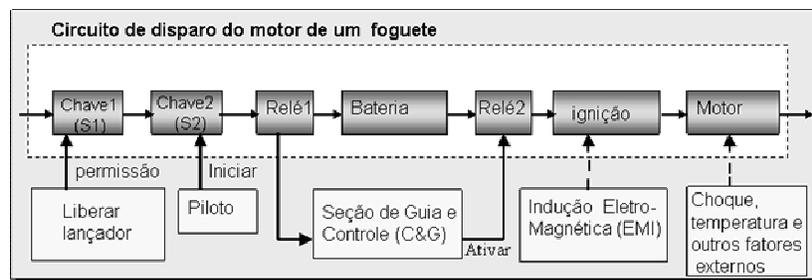


Figura 7.1 Diagrama de blocos de confiabilidade do circuito de disparo do motor.

O circuito de disparo, definido originalmente em [141] e mostrado na Figura 7.1, é projetado para permitir a partida do motor de acordo com as seguintes condições:

- 1- Abertura da chave S1, inicialmente fechada, de modo a permitir a liberação e o disparo do lançador;
- 2- Fechamento da chave de disparo S2 pelo Piloto para aplicação de potência ao Relé 1;
- 3- Ativação da Secção de Guia e Controle (C&G) pelo Relé 1;
- 4- Fechamento da ignição do circuito de disparo por meio do Relé 2, o qual é ativado pelo sinal de saída da Secção de Guia e Controle, para partida do motor do foguete.

O diagrama de blocos de confiabilidade da Figura 7.1 corresponde a eventos de sucesso, ou seja, eventos que uma vez ativados conduzem ao disparo do motor de forma confiável, enquanto eventos de falha podem ser representados por um modelo de árvore de falha. Através do modelo MDP, podem-se representar tanto um modelo quanto outro, bastando apenas inverter as transições imediatas correspondentes aos eventos de falha e de sucesso, e as regras lógicas dos circuitos seriais e paralelos. Os modelos de sucesso têm nos modelos de falha os seus duais.

O próximo passo consiste na construção dos modelos MDP capazes de representar os eventos de falha ou os modos de falha que conduzam o sistema a uma situação de disparo involuntário do motor, ou eventos de sucesso que disparem o motor em situações bem definidas. Uma vez desenvolvidos os modelos, deve-se atribuir a cada um deles as respectivas probabilidades de falha dos eventos ou dos modos de falha. As probabilidades dos modos de falha (X) para disparo inadvertido do motor são definidas na Tabela 7.1:

Tabela 7.1 Probabilidades de ocorrência dos modos de falha

Modo de Falha	Descrição	Probabilidade de Ocorrência
X _K	Chave S1 em curto	50 x 10 ⁻³
X _L	Chave S2 em curto	100 x 10 ⁻³
X _M	Fechamento indevido do Relé 1	40 x 10 ⁻³
X _N	Fechamento indevido do Relé 2	5 x 10 ⁻³
X _Q	Alta corrente de fuga	2 x 10 ⁻³
X _T	Transistor Q-2 aberto	1 x 10 ⁻³
X _U	Conector em curto para B+.	0,5 x 10 ⁻³

O sistema apresentado na Figura 7.1, originalmente desenvolvido por meio de árvore de falha, será desenvolvido neste estudo de caso por meio de modelos MDP, conforme descrição a seguir:

- 1) O circuito de disparo do motor será indevidamente habilitado se um dos modos de falha X_N, X_Q, X_T ou X_U ocorrer. Pode-se dizer ainda de um modo dual, que o circuito de disparo não liberará um sinal indevido de permissão para a seção de guia e

controle (G&C) se nenhum dos modos de falhas X_N , X_Q , X_T e X_U ocorrerem. Isto é representado pelo modelo MDP da Figura 7.2. Neste modelo cada um dos modos de falha é representado por um *token* ao qual uma correspondente probabilidade de falha é associada. A condição para não haver disparo involuntário, devido aos modos de falha, é definida por regras de decisão para uma configuração serial, conforme os *flags* da camada de configuração.

- 2) A bateria será indevidamente ativada se os modos de falhas X_K , X_L , e X_M ocorrerem simultaneamente. De uma forma dual, pode-se afirmar que a bateria permanecerá desativada se pelo menos um dos modos de falha X_K , X_L , ou X_M não ocorrer, conforme representado pela Figura 7.3. Neste modelo MDP cada modo de falha com sua respectiva probabilidade de ocorrência é representado por um *token*. A condição para não haver uma ativação inadvertida da bateria é definida pelos *flags* da camada de configuração, os quais definem regras de decisão para uma configuração paralela.

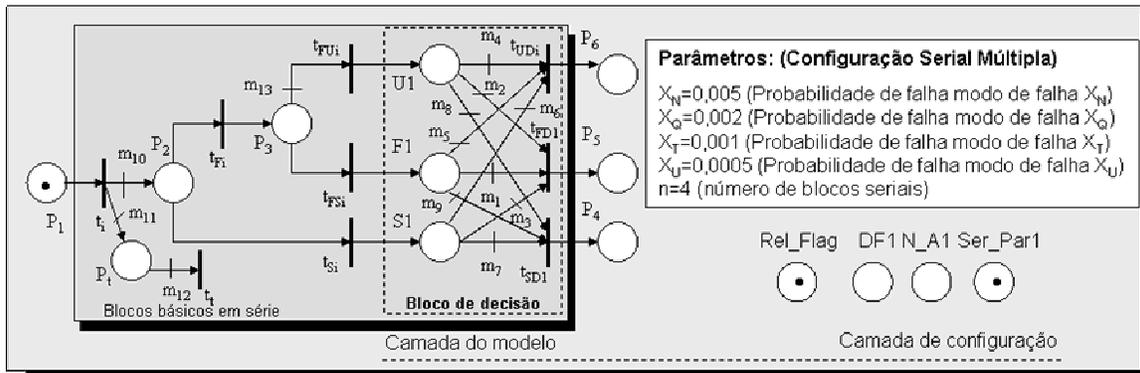


Figura 7.2 Modelo MDP para não habilitação do circuito de disparo.

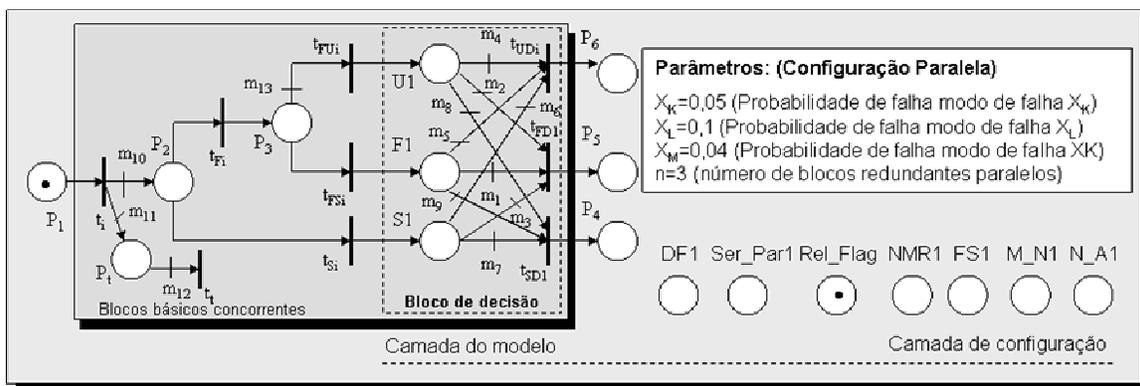


Figura 7.3 Modelo MDP para não ativação da bateria.

- 3) Haverá um disparo acidental do motor do foguete lançador de mísseis se houver uma ignição prematura devido a uma falha do hardware provocada pela ativação indevida da bateria e pela habilitação imprópria do circuito de disparo, ou devido a interferências eletromagnéticas, ou ainda devido a fatores externos como choques ou altas temperaturas do ambiente. O modelo dual, ou modelo de confiabilidade considera que o motor não disparará acidentalmente se não houver uma falha do hardware, nem interferências eletromagnéticas e nem causas externas causadas por

choques e altas temperaturas, conforme pode ser observado no modelo MDP da Figura 7.4. Neste modelo está representado o circuito de disparo do motor de um foguete lançador de mísseis conforme anteriormente mostrado na Figura 7.1 no formato de diagrama de blocos. O sub-modelo MDP da parte superior esquerda da Figura 7.4 representa o modelo da Figura 7.3 e o sub-modelo MDP da parte inferior esquerda representa o modelo da Figura 7.2. O sub-modelo ao centro representa a possibilidade de falha do hardware provocada pelos modos de falha dos modelos MDP das Figuras 7.2 e 7.3, anteriormente descritos. O sub-modelo MDP no centro a direita representa a ocorrência dos demais eventos de falha que levam o motor a um disparo involuntário, tais como as interferências eletromagnéticas cuja probabilidade de falha é de 5×10^{-6} , as altas temperaturas do ambiente cuja probabilidade de falha é de $2,5 \times 10^{-6}$ e os choques cuja probabilidade de falha é de $12,5 \times 10^{-6}$. O modelo MDP da Figura 7.4 foi desenvolvido para modelar a impossibilidade de disparo acidental do motor do foguete lançador de mísseis, embora estimativas de disparo acidental sejam obtidas na condição de falha do modelo. Contudo, o modelo MDP pode ser desenvolvido para modelar o disparo acidental bastando para isso construir o modelo dual. O modelo MDP, conforme mostrado na Figura 7.4, permite análise de sensibilidade sobre os parâmetros do modelo.

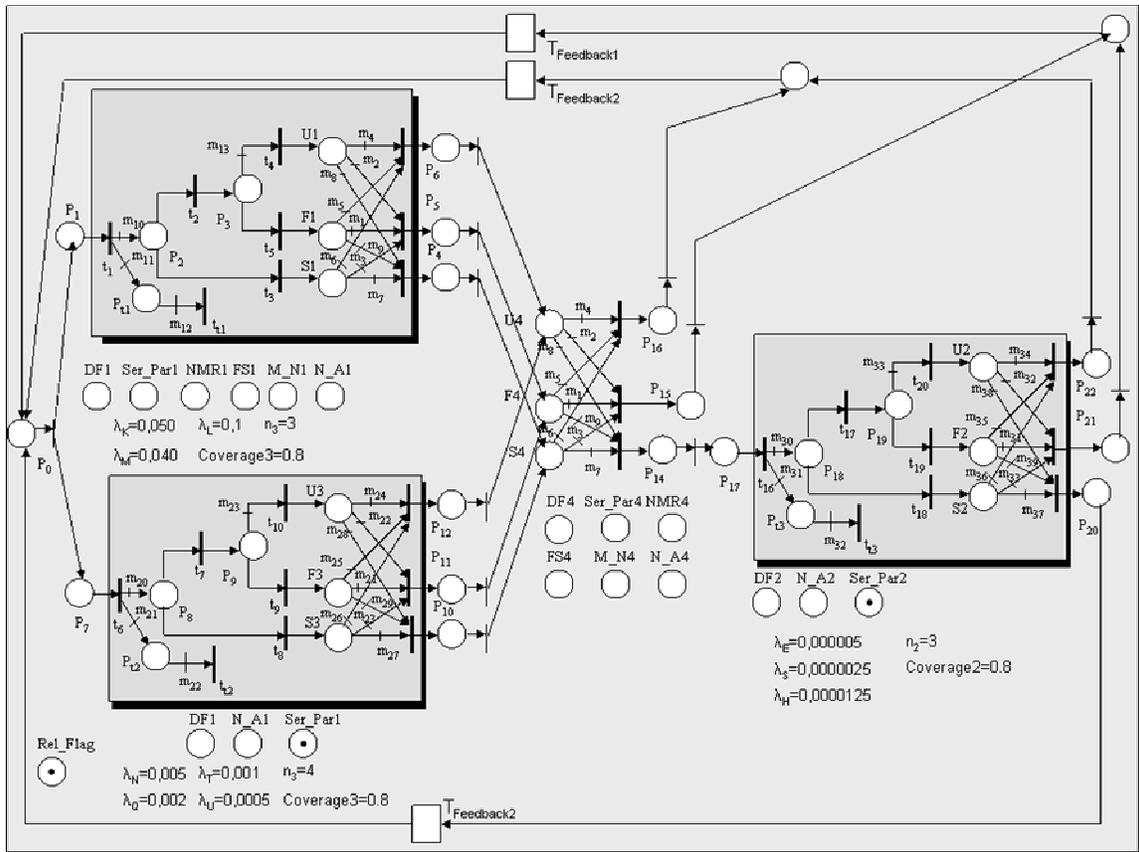


Figura 7.4 Modelo MDP do circuito de disparo do motor do foguete.

Alguns cenários podem ser construídos com respeito ao número de disparos acidentais. Pode-se observar que alguns eventos contribuem para uma maior ocorrência

de disparo acidental do motor, como por exemplo, as altas temperaturas e as interferências eletromagnéticas. Constatou-se que uma melhor isolamento térmica capaz de reduzir a probabilidade de ocorrência de altas temperaturas por um fator de 10, provoca uma diminuição da possibilidade de disparo indevido do motor em 52,17%, menos da metade, isto é, de 23 disparos/milhão para 11 disparos/milhão. Ou ainda, uma melhor blindagem capaz de reduzir as interferências eletromagnéticas, em especial as radiações infravermelhas por um fator de 10, ocasiona uma redução dos disparos indevidos do motor em torno de 21,7%, isto é, de 23 disparos/milhão para 18 disparos por milhão. O modelo MDP permite além destas análises de sensibilidade, análises referentes ao fator de cobertura, isto é, a análise de segurança do motor para diferentes probabilidades de detecção de falha. As estimativas obtidas não apenas corroboram com os valores obtidos nos modelos originais em [141] e desenvolvidos por meio de modelos de árvores de falha, como acrescentam as probabilidades de falhas seguras e inseguras resultantes dos fatores de cobertura de falhas, não contemplados no documento original. Os modelos MDP, conforme pôde ser visto, podem representar as técnicas de análise de dependabilidade por meio de diagramas de blocos de confiabilidade (RDB) ou por árvores de falha (FT). A metodologia proposta apresenta uma grande flexibilidade. Ela permite não apenas a definição de valores de probabilidade para os modos de falha e os eventos de falha, como também a definição funções de distribuição associadas a falhas e reparos, além de fatores de cobertura, os quais indicam a probabilidade de falhas serem detectadas. Atributos de confiabilidade, disponibilidade e segurança podem ser obtidos, através da metodologia proposta, sem que ocorra explosão de estados e *stiffness*.

7.2 Estudo de Caso II: Sistema de telecomunicações em suporte a um sistema de transmissão de energia elétrica

O presente estudo de caso tem como objetivo validar a metodologia desenvolvida nesta Tese, pela sua aplicação à modelagem e avaliação dos atributos de confiabilidade e disponibilidade da infra-estrutura de comunicação da automação da transmissão de energia elétrica, de uma grande empresa do setor, originalmente modelados e avaliados por meio da metodologia definida em [55]. No projeto original, denominado R-TOOL, foram definidas uma metodologia e modelos básicos, muitos dos quais utilizando técnicas de tolerância a falhas, de forma a modelar e avaliar a infra-estrutura de comunicação, de modo a satisfazer critérios de qualidade dos serviços, impostos por entidade reguladora do setor. Foram definidos também os passos necessários à avaliação dos atributos de dependabilidade, iniciando-se com a definição do problema, e prosseguindo-se com a escolha de critérios de modelagem e avaliação do sistema, até a análise dos resultados das avaliações realizadas.

Uma série de atividades foram desenvolvidas ao longo do projeto R-Tool: levantamento da infra-estrutura de comunicação, estudo das propriedades qualitativas, levantamento dos registros históricos de falhas e reparos para análise estatística, levantamento das características de dependabilidade dos equipamentos, fornecidas pelos fabricantes, estudo das técnicas de avaliação e definição do processo de escolha da

técnica de avaliação apropriada, dentre outras. O projeto foi realizado durante o período de 1 ano e os seus resultados foram considerados muito satisfatórios [55].

O estudo de caso a ser descrito, avalia os atributos de confiabilidade e de disponibilidade para dois dos trechos representativos da infra-estrutura de comunicação, através da metodologia aqui apresentada. Os trechos selecionados, denominados trecho I e trecho II, compõem um subsistema de telecomunicações, o qual é composto por diversos equipamentos, os quais por sua vez são compostos por vários dispositivos, cada qual formado por um único componente ou por um par de componentes, um primário e um outro redundante. Os equipamentos dos trechos I e II são semelhantes e estão disponibilizados em locais distintos, porém adjacentes. Apesar dos trechos selecionados terem uma quantidade limitada de equipamentos, a quantidade de estados gerados em um modelo EDSPN pode ser elevada, o que pode tornar impraticável a avaliação manual, sem o emprego de ferramentas de análise adequadas. A utilização das redes de Petri estocásticas como ferramenta de modelagem e avaliação, torna os modelos mais concisos, permitindo ao modelador uma visão sistêmica mais clara e abrangente. Na metodologia proposta, a representação do subsistema é feita de uma forma modular e hierárquica, inicialmente por meio de diagrama de blocos e em seguida por redes de Petri. O diagrama de blocos representativo de cada um dos trechos do subsistema de telecomunicações é formado por blocos correspondentes aos equipamentos dispostos na Figura 7.5. Cada bloco presente na Figura 7.5 é composto por vários outros blocos representando dispositivos, cada qual com um ou dois componentes, cuja interligação define a estrutura dos dispositivos dentro de cada equipamento, conforme a Figura 7.6.

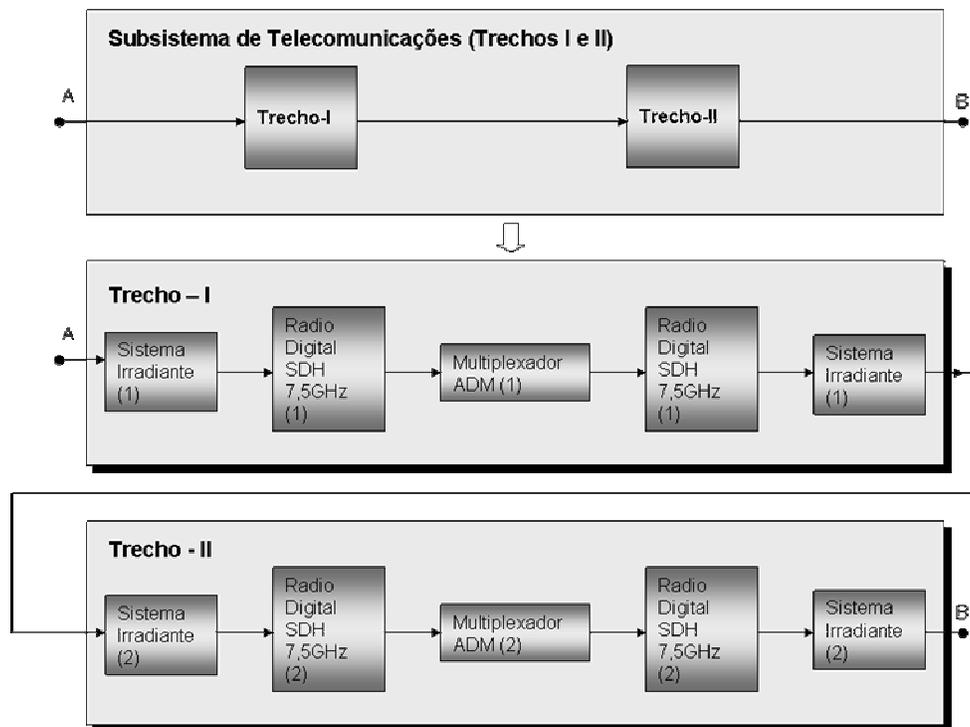


Figura 7.5 Diagrama de blocos hierárquico e modular da infra-estrutura de comunicação

De acordo com a metodologia proposta, cada bloco de dispositivo é transformado no correspondente modelo EDSPN, para obtenção das expressões analíticas ou numéricas, as quais serão utilizadas no nível hierárquico seguinte, composto pelos modelos MDP. Os tempos de vida dos dispositivos que compõem os equipamentos, no diagrama de blocos da Figura 7.5, são exponencialmente distribuídos com média $1/\lambda_i$, onde λ_i é a taxa de falha constante do dispositivo i . Os tempos de reparo por sua vez, também exponencialmente distribuídos, podem apresentar duas taxas distintas, uma vez que aspectos de detecção de falha podem ou não serem considerados. Caso a cobertura de falhas seja considerada perfeita, o tempo médio de reparo é dado por $1/\mu_i$; caso a cobertura de falhas não seja perfeita o tempo médio de reparo das falhas detectadas é dada por $1/\mu_i$, enquanto o tempo médio das falhas não detectadas é a soma do tempo médio de reparo propriamente dito, mais um tempo, bastante longo, de percepção da ocorrência de falha, denominado de tempo médio de percepção, o qual é considerado exponencialmente distribuído com média $1/\mu_j$, onde μ_j é a taxa de percepção do erro do dispositivo i . Quando a cobertura de falha é imperfeita o processo de reparo segue uma distribuição hiperexponencial, formada pelas distribuições exponenciais μ_i ; e μ_j , e pelas probabilidades de detecção (CF_i) e de não detecção das falhas ($1 - CF_i$), respectivamente. Os tempos médios, representados pelo inverso das taxas, são utilizados nas expressões dos modelos MDP correspondente a cada componente dos dispositivos.

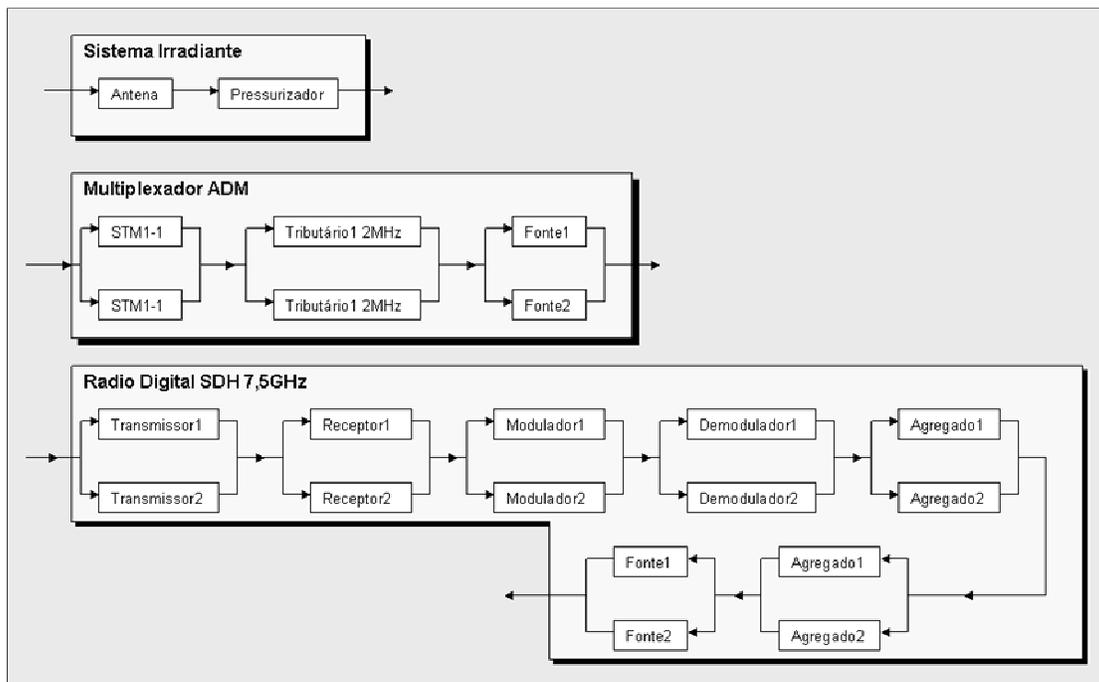


Figura 7.6 Diagrama de blocos dos equipamentos do subsistema de telecomunicações.

Os blocos que representam os dispositivos podem operar com ou sem redundância. Quando os dispositivos de um equipamento estão ativos e em operação simultânea, o modelo MDP de bloco ativo é utilizado; quando os dispositivos de um equipamento estão em série, pode-se representar os dispositivos por meio de um modelo

MDP de blocos seriais múltiplos, conforme descrito no Capítulo 5. Na Tabela 7.2 alguns dos parâmetros associados a cada um dos dispositivos são apresentados.

Tabela 7.2 Definição dos parâmetros dos diversos dispositivos dos trechos I e II

Equip.	Dispositivos	Componente Ativo+ (redundante)	MTBF	MTTF	M T T R	Número de estados	
						CF=1	CF≠1
Sistema Irradiante	Antena	1	350000	349992	8	02	03
	Pressurizador	1	350000	349992	8	02	03
Rádio Digital SDH 7.5GHz	Transmissor	1 + (1)	172800	172792	8	03	06
	Receptor	1 + (1)	172800	172792	8	03	06
	Modulador	1 + (1)	155520	155512	8	03	06
	Demodulador	1 + (1)	155520	155512	8	03	06
	Placa Agregada	1 + (1)	276480	276472	8	03	06
	Placa Tributário	1 + (1)	276480	276472	8	03	06
	Fonte	1 + (1)	657000	656992	8	03	06
Mux ADM	STM1 Agregado	1 + (1)	248138	248130	8	03	06
	Tributário 2MHz	1 + (1)	271076	271068	8	03	06
	Fonte	1 + (1)	657000	656992	8	03	06

Assume-se que os dispositivos que compõem cada equipamento são estocasticamente independentes, com distribuições exponenciais associadas a falhas e reparos .

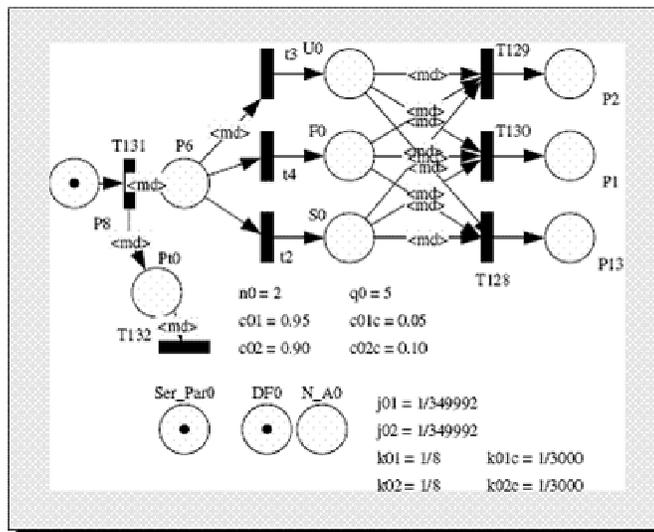


Figura 7.7 Modelo MDP correspondente ao Sistema Irradiante

Na metodologia original do projeto R-Tool, cada dispositivo é analisado isoladamente, sendo o número de estados de cada equipamento, proporcional ao produto do número de estados gerados por cada dispositivo, o que pode resultar numa grande quantidade de estados. Por outro lado, as representações por meio de modelos MDP, dos equipamentos que compõem cada trecho, não apresentam este problema devido ao seu espaço de

estados reduzido. Nas Figuras 7.7, 7.8 e 7.9 são mostrados os modelos MDP, correspondentes aos equipamentos sistema irradiante, mux ADM e rádio digital SDH 7.5GHz, respectivamente, da forma como eles foram utilizados na ferramenta de modelagem.

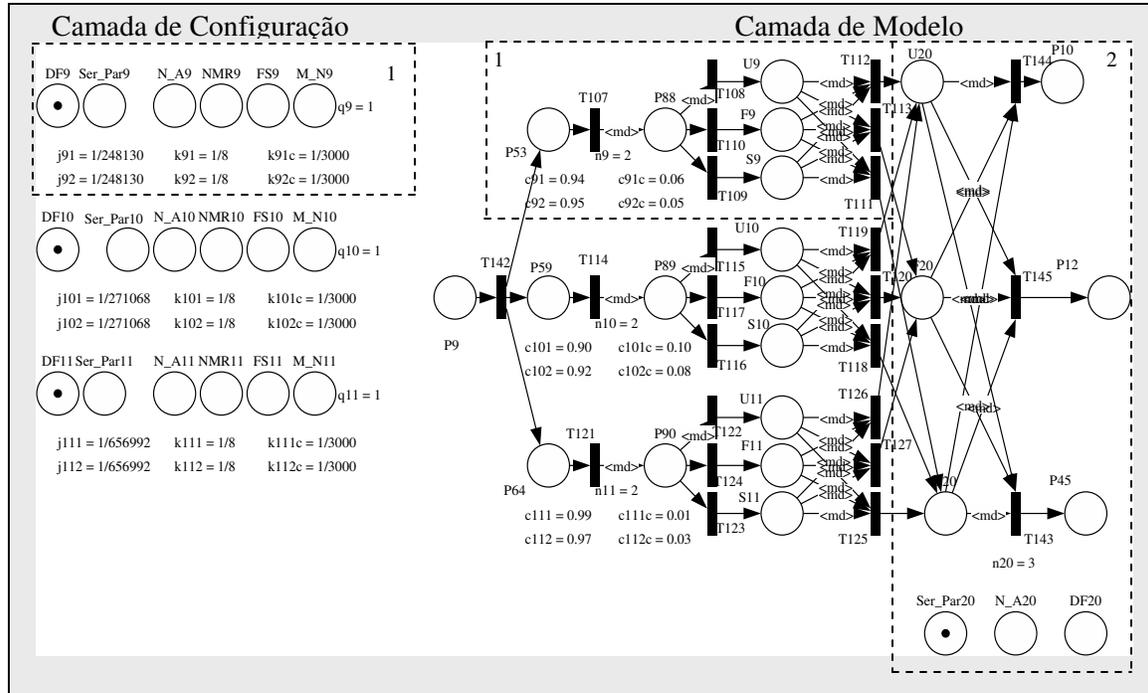


Figura 7.8 Modelo MDP correspondente ao Multiplexador ADM

Cada um dos modelos apresentados é composto por duas camadas: camada de modelo e camada de configuração (conjunto de lugares isolados). A camada de modelo representa, por exemplo, a estrutura de interligação dos dispositivos nos equipamentos, enquanto a camada de configuração define as diversas configurações possíveis para a camada de modelo, ou seja, as diversas formas de configuração desses dispositivos nos equipamentos. Nos modelos MDP, são definidos o tipo de configuração, as taxas de falha e de reparo de cada dispositivo, no caso das coberturas de falhas serem perfeitas, ou as taxas de falha, de reparo e de percepção de cada dispositivo, no caso da cobertura de falhas serem imperfeitas, com possibilidade de diferentes taxas para o componente primário e para o componente redundante. Também são definidos o número de componentes em cada dispositivo e as quantidades de dispositivos em cada equipamento, além das coberturas de falhas dos componentes de cada dispositivo. O intervalo de tempo considerado para análise transiente é de 1 ano ou 8760 horas. O modelo MDP de cada trecho do subsistema de telecomunicações é mostrado na Figura 7.10. O equipamento sistema irradiante mostrado na Figura 7.7, por exemplo, apresenta os seguintes parâmetros de configuração: $n_0=2$ (número de dispositivos do sistema irradiante), representa o número de *tokens* a ser colocado no lugar P6 pelo disparo da transição imediata T131; c_{0x} e c_{0xc} , para $x=1,2$, representam as probabilidades de detecção e não-detecção de falhas, respectivamente, para os dispositivos 1(antena) e 2(pressurizador);

$j0x$, para $x=1,2$, representa as taxas de falha dos dispositivos 1 e 2; $k0x$, para $x=1,2$, representa as taxas de reparo dos dispositivos 1 e 2; $k0xc$, para $x=1,2$, representa as taxas de percepção do erro dos dispositivos 1 e 2. A Figura 7.8 representa o equipamento *mux* ADM, o qual é composto por três dispositivos, cada qual com dois componentes redundantes em paralelo. Retângulos pontilhados correspondentes a um dos dispositivos e ao mecanismo de decisão, com os números 1 e 2, respectivamente, na parte de cima, foram colocados para facilitar o entendimento. O dispositivo representado pelo pontilhado 1, é composto por dois componentes ($n9=2$), cada um dos quais com suas respectivas taxas de falha ($j91=1/248130$ e $j92=1/248130$), taxas de reparo ($k91=1/8$ e $k92=1/8$), taxas de percepção ($k91c=1/3000$ e $k92c=1/3000$) e fatores de cobertura ($c91=0.94$ e $c92=0.95$, correspondentes a detecção de falhas, e $c91c=0.06$ e $c92c=0.05$, correspondentes a não detecção de falhas). Os *flags* #Ser_Par9=0 (ausência de *token* no lugar Ser_Par9) e #DF9=1 (presença de *token* no lugar DF9) indicam que os dois componentes estão em paralelo e com o mecanismo de detecção de falha ativado. O mecanismo de decisão do equipamento indica que os três dispositivos que compõem o equipamento *mux* ADM ($n20=3$), estão em série, conforme indicado pelo *flag* #Ser_Par20=1. Caso os três dispositivos do equipamento *mux* ADM estejam operacionais, um *token* será colocado no lugar P45. Caso contrário um *token* será colocado no lugar P12, correspondente a falha segura, ou P10, correspondente a falha insegura ou catastrófica. As regras relativas ao paralelismo e a serialização de componentes estão definidas no capítulo 5 e a sintaxe correspondente a ferramenta de modelagem está descrita no ANEXO-B. Os demais dispositivos da Figura 7.8 e os dispositivos da Figura 7.9, têm a mesma explicação do dispositivo anterior.

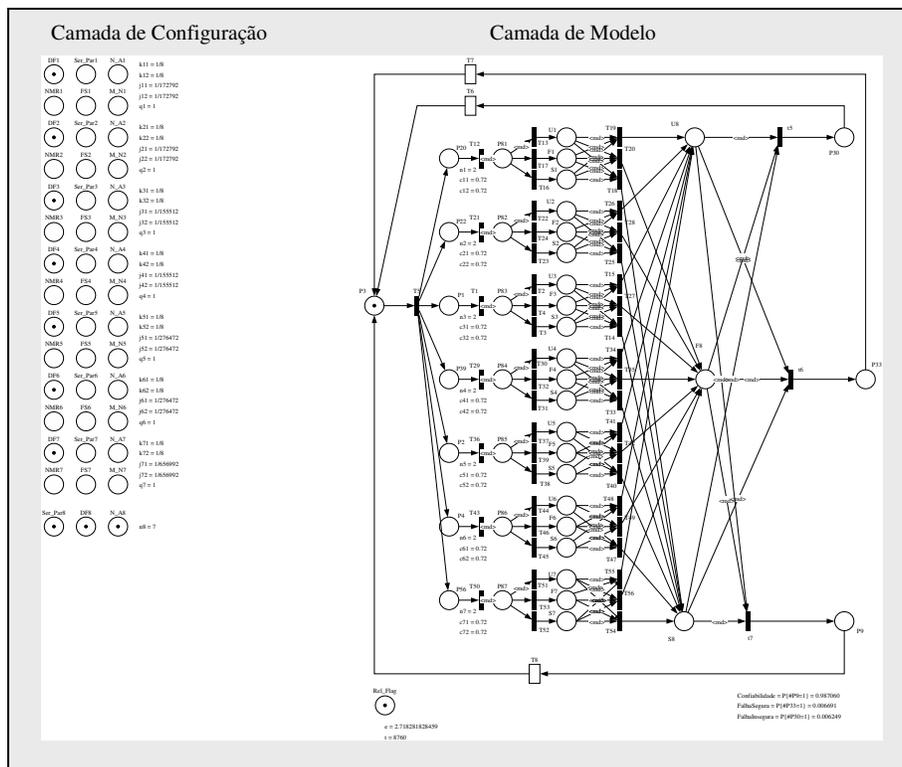


Figura 7.9 Modelo MDP correspondente ao Radio Digital SDH 7,5GHz

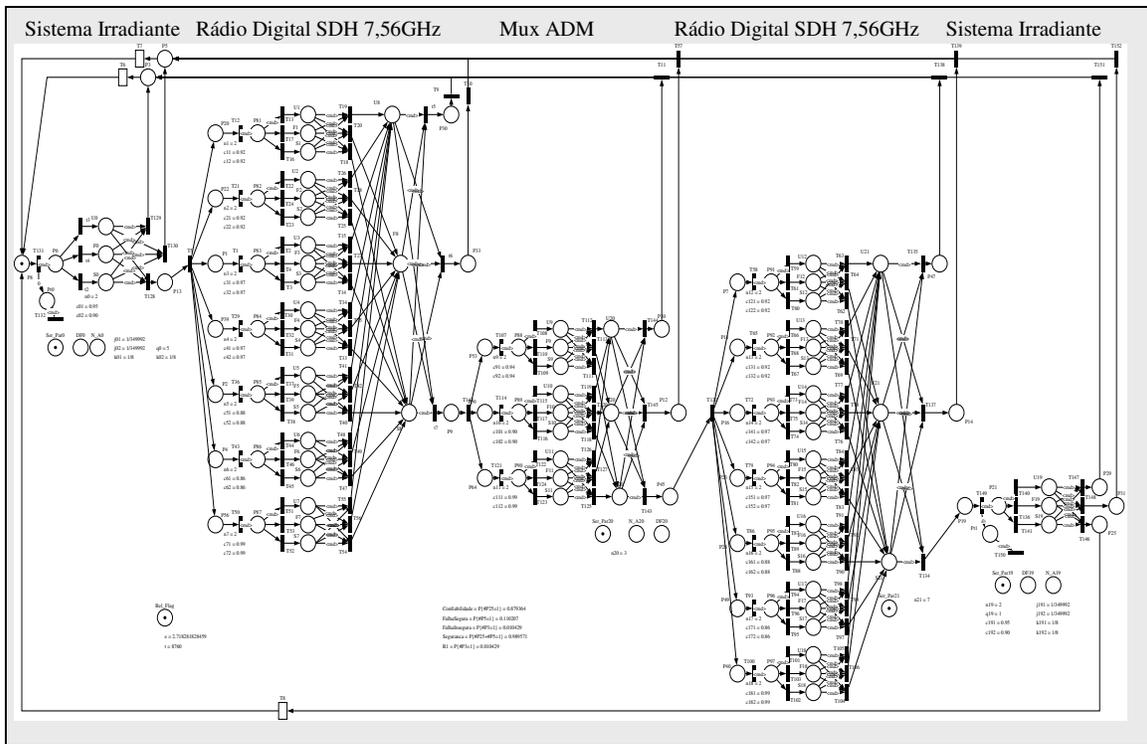


Figura 7.10 Modelo MDP correspondente aos Trechos I ou II

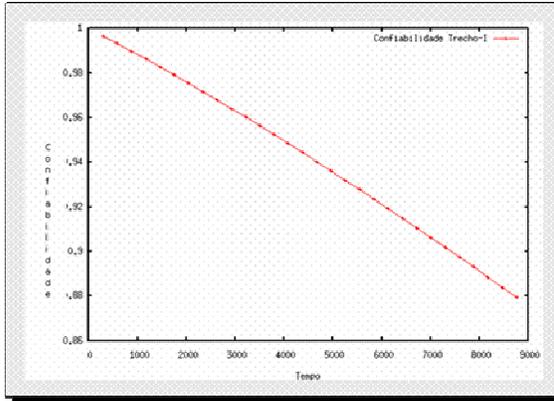
Na Figura 7.10 é dada uma visão geral dos modelos de todos os equipamentos que compõem o trecho I (ou trecho II), sendo omitidos vários *flags* das camadas de configuração, para não sobrecarregar a imagem. Na ferramenta de modelagem, contudo, a análise é feita considerando-se todos os *flags* das camadas de configuração. Na Figura 7.10 os modelos dos dispositivos que compõem cada um dos equipamentos são mostrados de um modo macro. A explicação dos dispositivos da Figura 7.10 é a mesma observada para os dispositivos da Figura 7.8, correspondente ao Mux ADM. No presente estudo de caso são considerados dois cenários distintos. No primeiro cenário é admitida uma cobertura de falhas perfeita para o modelo MDP, enquanto no segundo cenário admite-se uma cobertura de falhas imperfeita (o segundo cenário não foi utilizado na metodologia original empregada na empresa de energia elétrica). As medidas de confiabilidade e de disponibilidade para o primeiro cenário são mostradas nas Tabelas 7.3 e 7.4, respectivamente, considerando-se um intervalo de tempo entre pontos de verificação, IT, de 292 horas para o grafo transiente. Um grafo de confiabilidade transiente com 30 pontos de verificação é apresentado na Figura 7.11. Este grafo é construído por meio de uma sequência de análises de estado permanente, uma para cada ponto de verificação. Caso haja necessidade de uma maior precisão para o grafo de confiabilidade um maior número de pontos deve ser levado em conta.

Tabela 7.3 Confiabilidade com cobertura perfeita de falhas

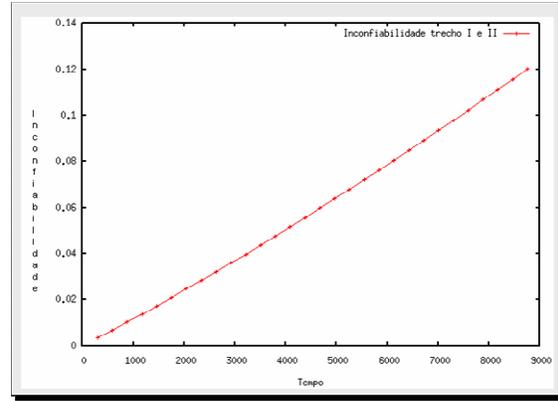
Pontos de Verificação	Tempo em Horas	Confiabilidade (em %)	Inconfiabilidade (em %)
01	292	0.99663538	0.00336462
02	584	0.99321660	0.00678340
03	876	0.98974460	0.01025540
04	1168	0.98622035	0.01377965
05	1460	0.98264478	0.01735522
06	1752	0.97901886	0.02098114
07	2044	0.97534355	0.02465645
08	2336	0.97161979	0.02838021
09	2628	0.96784855	0.03215145
10	2920	0.96403078	0.03596922
11	3212	0.96016744	0.03983256
12	3504	0.95625948	0.04374052
13	3796	0.95230785	0.04769215
14	4088	0.94831351	0.05168649
15	4380	0.94427741	0.05572259
16	4672	0.94020050	0.05979950
17	4964	0.93608373	0.06391627
18	5256	0.93192803	0.06807197
19	5548	0.92773436	0.07226564
20	5840	0.92350365	0.07649635
21	6132	0.91923685	0.08076315
22	6424	0.91493488	0.08506512
23	6716	0.91059868	0.08940132
24	7008	0.90622917	0.09377083
25	7300	0.90182728	0.09817272
26	7592	0.89739393	0.10260607
27	7884	0.89293003	0.10706997
28	8176	0.88843650	0.11156350
29	8468	0.88391423	0.11608577
30	8760	0.87936413	0.12063587

Tabela 7.4 Disponibilidade com cobertura perfeita de falhas

Atributo	Probabilidade (em %)
Disponibilidade	0.99990855
Indisponibilidade	0.00009145



(a) Confiabilidade do trecho I (II)

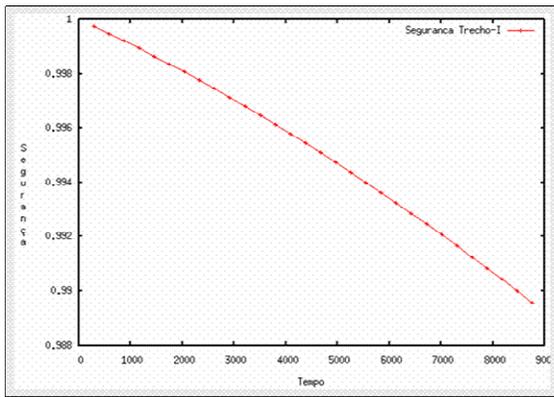


(b) Inconfiabilidade do trecho I (II)

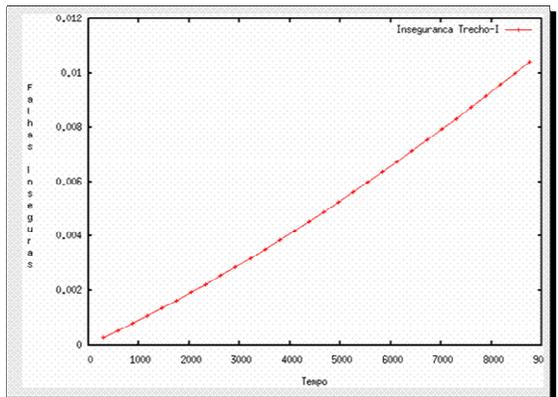
Figura 7.11 Gráficos relativos ao trecho I (II) do subsistema com cobertura perfeita.

Tabela 7.5 Fatores de cobertura dos componentes

Dispositivo	Fatores de Cobertura
Antena	0.95
Pressurizador	0.90
Transmissor	0.92
Receptor	0.97
Modulador	0.88
Demodulador	0.86
Agregado	0.94
Tributário	0.90
Fonte	0.99



(a) Segurança do trecho I (II)



(b) Falhas inseguras do trecho I (II)

Figura 7.12 Gráficos relativos ao trecho I (II) do subsistema com cobertura imperfeita.

Para o segundo cenário, são arbitrados diversos fatores de cobertura para os diversos componentes dos dispositivos, de acordo com a Tabela 7.5. Com isto, pode-se obter além das medidas de confiabilidade e disponibilidade, medidas de segurança. A Tabela 7.6, correspondente ao segundo cenário, é composta pelos valores de confiabilidade, falha segura, falha insegura e segurança, enquanto os valores de disponibilidade são mostrados na Tabela 7.7. Na Figura 7.12 são apresentados os gráficos

de segurança e de falhas inseguras. O gráfico de confiabilidade para o segundo cenário é semelhante ao gráfico de confiabilidade do primeiro cenário, mostrado na Figura 7.11a, enquanto o gráfico de inconfiabilidade do primeiro cenário, mostrado na Figura 7.11b, corresponde a soma das falhas seguras e inseguras do segundo cenário.

Tabela 7.6 Valores de dependabilidade para cobertura de falhas imperfeita

Ponto de Verificação	Tempo em Horas	Confiabilidade (em %)	Falha Segura (em %)	Falha Insegura (em %)	Segurança (em %)
01	292	0.99663538	0.00311060	0.00025402	0.99974598
02	584	0.99321660	0.00626797	0.00051543	0.99948457
03	876	0.98974460	0.00947126	0.00078413	0.99921587
04	1168	0.98622035	0.01271963	0.00106002	0.99893998
05	1460	0.98264478	0.01601222	0.00134300	0.99865700
06	1752	0.97901886	0.01934817	0.00163297	0.99836703
07	2044	0.97534355	0.02272663	0.00192982	0.99807018
08	2336	0.97161979	0.02614674	0.00223347	0.99776653
09	2628	0.96784855	0.02960765	0.00254379	0.99745621
10	2920	0.96403078	0.03310851	0.00286071	0.99713929
11	3212	0.96016744	0.03664845	0.00318410	0.99681590
12	3504	0.95625948	0.04022663	0.00351389	0.99648611
13	3796	0.95230785	0.04384219	0.00384995	0.99615005
14	4088	0.94831351	0.04749428	0.00419221	0.99580779
15	4380	0.94427741	0.05118205	0.00454054	0.99545946
16	4672	0.94020050	0.05490464	0.00489486	0.99510514
17	4964	0.93608373	0.05866121	0.00525507	0.99474493
18	5256	0.93192803	0.06245091	0.00562106	0.99437894
19	5548	0.92773436	0.06627290	0.00599274	0.99400726
20	5840	0.92350365	0.07012634	0.00637001	0.99362999
21	6132	0.91923685	0.07401038	0.00675277	0.99324723
22	6424	0.91493488	0.07792420	0.00714092	0.99285908
23	6716	0.91059868	0.08186696	0.00753436	0.99246564
24	7008	0.90622917	0.08583783	0.00793300	0.99206700
25	7300	0.90182728	0.08983598	0.00833674	0.99166326
26	7592	0.89739393	0.09386059	0.00874548	0.99125452
27	7884	0.89293003	0.09791084	0.00915913	0.99084087
28	8176	0.88843650	0.10198592	0.00957759	0.99042241
29	8468	0.88391423	0.10608501	0.01000076	0.98999924
30	8760	0.87936413	0.11020732	0.01042855	0.98957145

Tabela 7.7 Disponibilidade com cobertura perfeita de falhas

Atributo	Probabilidade (em %)
Disponibilidade	0.99860897
Indisponibilidade	0.00139103

A representação do subsistema, formado pelos trechos I e II, é composta por dois modelos MDP de subsistemas, um para cada trecho, os quais apresentam expressões numéricas formadas por 30 termos correspondentes aos 30 pontos de verificação de confiabilidade, falha segura e falha insegura, além do valor de disponibilidade.

Na Figura 7.13 é mostrado o modelo MDP de subsistema formado pelos modelos MDP correspondentes aos trechos I e II. Na Figura 7.14 são mostrados os gráficos de confiabilidade e inconfiabilidade do subsistema. Na Tabela 7.8 são descritas as medidas de confiabilidade e de inconfiabilidade para os pontos de validação do subsistema, considerando-se que todos os componentes de todos os dispositivos têm cobertura perfeita. Na Tabela 7.9 é apresentada o valor de disponibilidade do subsistema

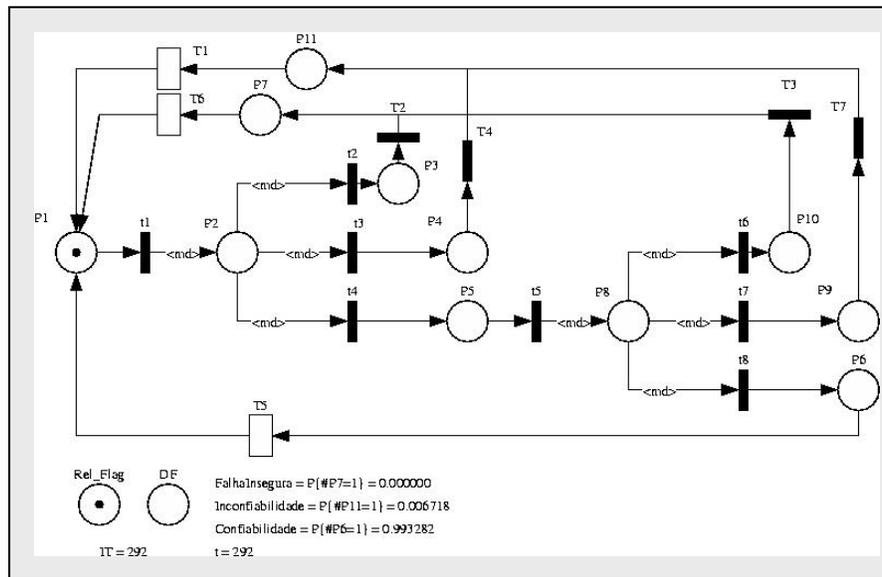
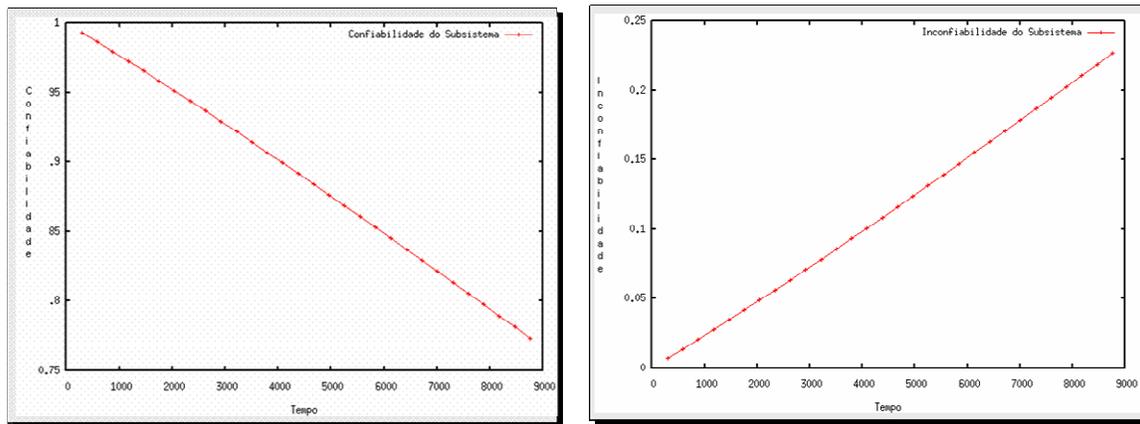


Figura 7.13 Modelo MDP do subsistema formado pelos trechos I e II.



(a) Confiabilidade do Subsistema

(b) Inconfiabilidade do Subsistema

Figura 7.14 Gráficos relativos ao subsistema com cobertura perfeita.

Tabela 7.8 Confiabilidade do subsistema com cobertura perfeita de falhas

Ponto de Verificação	Tempo em Horas	Confiabilidade (em %)	Inconfiabilidade (em %)
01	292	0.99328208	0.00671792
02	584	0.98647921	0.01352079
03	876	0.97959437	0.02040563
04	1168	0.97263058	0.02736942
05	1460	0.96559076	0.03440924
06	1752	0.95847793	0.04152207
07	2044	0.95129504	0.04870496
08	2336	0.94404502	0.05595498
09	2628	0.93673082	0.06326918
10	2920	0.92935534	0.07064466
11	3212	0.92192151	0.07807849
12	3504	0.91443219	0.08556781
13	3796	0.90689024	0.09310976
14	4088	0.89929851	0.10070149
15	4380	0.89165983	0.10834017
16	4672	0.88397698	0.11602302
17	4964	0.87625275	0.12374725
18	5256	0.86848985	0.13151015
19	5548	0.86069104	0.13930896
20	5840	0.85285899	0.14714101
21	6132	0.84499639	0.15500361
22	6424	0.83710583	0.16289417
23	6716	0.82918996	0.17081004
24	7008	0.82125131	0.17874869
25	7300	0.81329244	0.18670756
26	7592	0.80531587	0.19468413
27	7884	0.79732404	0.20267596
28	8176	0.78931941	0.21068059
29	8468	0.78130437	0.21869563
30	8760	0.77328127	0.22671873

Tabela 7.9 Disponibilidade do subsistema

Atributo	Probabilidade (em %)
Disponibilidade	0.99981711
Indisponibilidade	0.00018289

Pode-se observar dos resultados obtidos, que a incorporação de novos trechos ao subsistema de telecomunicações degrada as medidas de dependabilidade. Um outro fator importante são as coberturas de falha dos componentes. Quanto menor a capacidade de detecção de falhas piores serão as características de dependabilidade. Para que o sistema atinja metas de disponibilidade e/ou confiabilidade é necessário a incorporação de

técnicas de tolerância a falhas aos dispositivos por meio de componentes redundantes ou mesmo a criação de rotas alternativas, no caso das redes de telecomunicações.

Este estudo de caso mostra que os critérios de dependabilidade avaliados por meio da metodologia aqui proposta, através de tabelas e gráficos, correspondem com precisão, aos resultados obtidos pela metodologia adotada no projeto R-Tool. Nesta metodologia, porém, as características de confiabilidade e disponibilidade são todas obtidas por meio de análise transiente e de estado permanente, diferentemente da metodologia original, onde alguns resultados são obtidos por simulação, devido a dimensão do espaço de estados. O tempo gasto pela ferramenta de modelagem, para análise dos modelos, é bastante reduzido, uma vez que existem apenas três marcações tangíveis para qualquer modelo MDP, representando as condições sucesso, falha segura e falha insegura. Além disso, esta metodologia permite a análise dos atributos de segurança, não demonstrados na metodologia original e permite que os parâmetros MTTF, MTTR, MTEP e fator de cobertura dos componentes de qualquer dispositivo, possam assumir qualquer valor, diferentemente da metodologia original, onde estes parâmetros possuem os mesmos valores para os componentes de um mesmo dispositivo. A metodologia proposta apresenta como características marcantes, a reduzida geração do espaço de estados e a grande flexibilidade por meio da parametrização.

7.3 Estudo de Caso III: sistema de controle de vôo de aeronaves

O presente estudo de caso descreve o desenvolvimento de arquiteturas candidatas dos sistemas de controle de vôo de aeronaves, as quais foram originalmente propostas em [70], para análise de confiabilidade. Nesse estudo de caso, os principais componentes de um sistema de controle de vôo são apresentados e configurados, conforme as arquiteturas propostas, e medidas não apenas de confiabilidade, mas de disponibilidade e de segurança são obtidas. Para a obtenção de medidas de disponibilidade, considera-se que todas as ocorrências de falhas são detectadas, ou seja, que a cobertura de falhas é perfeita.

A estrutura básica dos dispositivos eletrônicos de uma aeronave que utiliza tecnologia de controle *fly-by-wire*, é mostrada na Figura 7.15. Nesta figura, observa-se que o sistema de computador central, baseado nos comandos ativados pela tripulação, posiciona os dispositivos de controle aerodinâmicos adequadamente de modo a manter um controle efetivo da aeronave. Adicionalmente, o sistema de computador central comunica através de *displays*, dispostos na cabine de comando, informações de desempenho do motor, dos pontos de verificação do plano de vôo, dentre outros, além de controlar a comunicação, a navegação, e o sistema de advertência em caso de anomalias.

De um modo geral, o sistema de controle da aeronave é subdividido em dois subsistemas: um sistema de controle de vôo e um sistema de gerenciamento de vôo. As arquiteturas do sistema de gerenciamento de vôo, compostas pelos Módulos 2 e 5 da Figura 7.15, suportam o uso dos equipamentos de comunicação, navegação e advertência, enquanto as arquiteturas dos sistemas de controle de vôo, compostas pelos Módulos 1,3 e 4, suportam os dispositivos mecânicos e eletrônicos que fazem com que a aeronave se

mantenha em equilíbrio no ar, através do controle da velocidade e das superfícies aerodinâmicas. Neste estudo de caso as soluções a serem analisadas dizem respeito apenas as arquiteturas candidatas do sistema de controle de voo, devido as graves conseqüências que poderão advir em caso de eventuais falhas deste sistema.

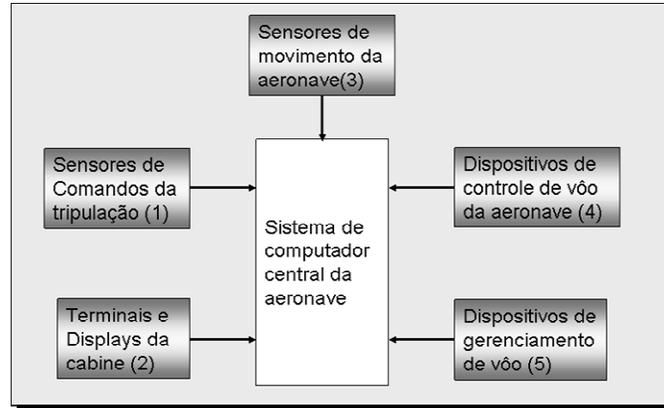


Figura 7.15 Sistema de controle da aeronave

O sistema de controle de voo é constituído por componentes mecânicos e eletrônicos cuja correta operação através de um determinado tempo de missão, garante que a aeronave possa voar com segurança. É necessário, portanto, que se tenha uma alta confiabilidade apenas durante o intervalo de tempo correspondente ao tempo de voo. De modo a minimizar a transferência dos dados sobre os meios de comunicação, os quais estão sujeitos a vários tipos de interferências, e de modo a particionar o sistema em blocos menores de modo a facilitar o projeto, arquiteturas de processamento distribuído são consideradas.

A configuração funcional do sistema distribuído projetado leva em conta somente o controle dos dispositivos que compõem a estrutura dinâmica da aeronave, ou seja, *aileron*s, *flaps*, *rudder* e *elevator*s, de modo a simplificar o projeto, conforme a Figura 7.16. O objetivo é projetar a parte eletrônica do sistema de controle de voo, de modo a evitar que ela seja a parte menos confiável do sistema, cujos dispositivos atuadores, os quais são dispositivos mecânicos, têm uma confiabilidade estimada de 0.97 (0.9999999), por hipótese. Portanto, os dispositivos eletrônicos devem ser no mínimo tão confiáveis quanto os atuadores ao final do tempo de missão. Além disso, considera-se que o sistema deva ser capaz de suportar pelo menos uma falha sem que haja conseqüências desastrosas. Conseqüentemente, o sistema além de ser confiável deve ser tolerante a falhas. Considera-se ainda que todas as arquiteturas de projeto sejam baseadas no conceito fundamental de *flux-summing* de forma a permitir que seja produzido um único resultado sobre os atuadores.

Outra importante decisão de projeto relaciona-se à topologia da rede de comunicação para transferência de dados entre o computador central e os atuadores. A interconexão entre os computadores central e local é definida de acordo com as configurações das arquiteturas candidatas. Duas soluções de interconexão são

consideradas: a) interconexão por meio de barramentos redundantes; b) interconexão ponto-a-ponto.

Em resumo, as arquiteturas candidatas devem satisfazer as seguintes restrições:

- a primeira falha deve ser 100% tolerada;
- a confiabilidade das partes eletrônicas devem ser no mínimo igual aquela das partes mecânicas, ou seja, 9_7 ou superior, durante o tempo de vôo;
- o processamento deve ser fisicamente distribuído através da aeronave, com o computador central controlando as ações gerais e os processadores periféricos controlando os atuadores individuais próximos a eles.

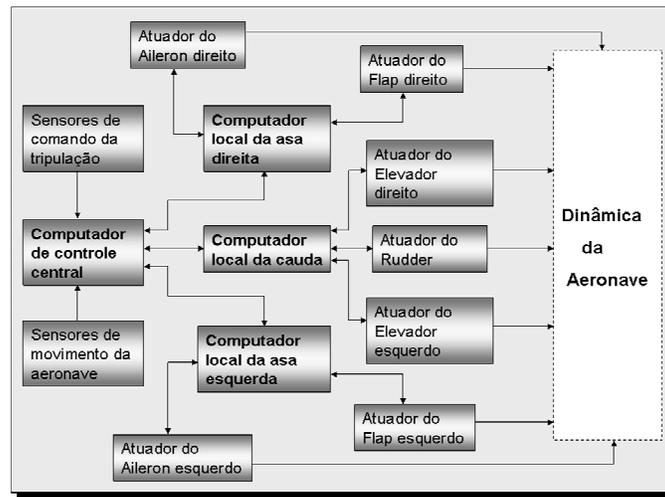


Figura 7.16 Sistema de controle de vôo

Como pode ser observado, a primeira restrição elimina todas as técnicas de tolerância a falhas que utilizam redundância dinâmica. Isto se constata, uma vez que para garantir que a primeira falha seja 100% detectável, a técnica de tolerância a falhas com redundância dinâmica, a ser utilizada, deve garantir que o mecanismo de cobertura de falhas seja 100% perfeito, o que não é possível se verificar. Portanto, após análise, verifica-se que as técnicas de tolerância a falhas com redundância estática são as que melhor se adequam, sendo escolhidas três delas para o projeto do sistema de controle de vôo. As técnicas escolhidas são:

- **técnica TTMR:** nesta solução de projeto, a técnica de tolerância a falhas TMR é utilizada para o computador central e para o barramento, de modo a garantir 100% de tolerância a falhas para primeira falha. Cada computador central tem um conjunto de sensores dedicados para a determinação do estado da aeronave e dos comandos ativados pela tripulação. O valor produzido por cada computador central é submetido a um mecanismo de votação, o qual é triplicado por segurança, e enviado a um barramento com redundância TMR. Os valores recebidos dos barramentos pelos computadores locais são votados novamente e em seguida processados. Os sinais de corrente para controle dos atuadores têm seus fluxos magnéticos somados de acordo com a técnica de *flux-summing*. Por meio de *flux-summing* é possível se tolerar até 2 falhas nos computadores locais,

desde que estas falhas sejam detectadas e o fluxo total continue a ser fornecido pelo computador local, ainda ativo. Esta arquitetura candidata utiliza um modelo TMR triplicado (TTMR) e uma estrutura de barramento conforme mostrado na Figura 7.17.

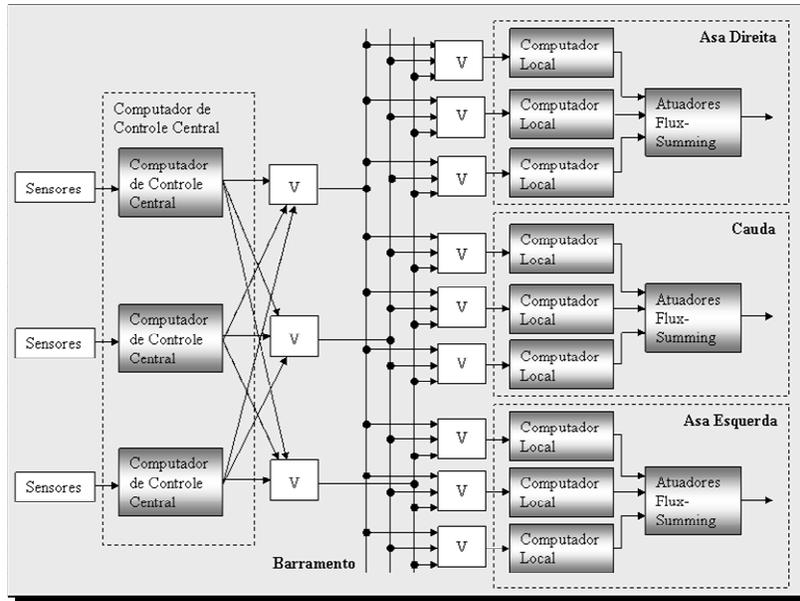


Figura 7.17 Arquitetura candidata TTMR

Esta arquitetura pode ser vista como uma conexão serial de cinco componentes TMR: um sistema de computador central em série com um sistema de barramento com redundância tripla e em série com três sistemas formados por computadores locais que controlam os dispositivos dispostos nas asas e na cauda da aeronave. A configuração da arquitetura TTMR é mostrada na Figura 7.18.

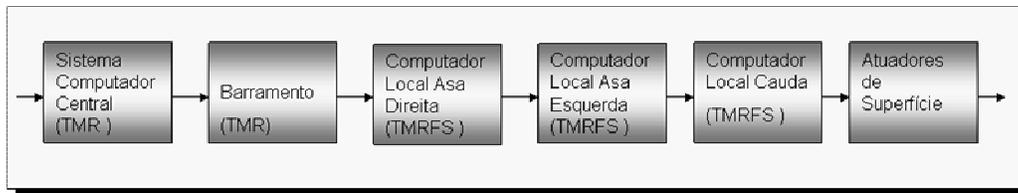


Figura 7.18 Arquitetura candidata TTMR – Configuração serial

Nas análises de confiabilidade, disponibilidade e segurança a serem apresentadas, consideram-se o conjunto constituído pelos votadores e barramento, como estando contido no bloco barramento da Figura 7.18, com uma configuração TMR. Os sistemas do computador central, do barramento e dos computadores locais são representados pelo modelo MDP da Figura 7.19.

Conforme pode ser observado na Figura 7.19, a camada de modelo, para a configuração TMR do bloco de sistema do computador central e do barramento, é semelhante a camada de modelo das configurações TMR com *flux-summing* (FS) dos blocos dos computadores locais. A única diferença entre elas corresponde a camada de configuração. A Tabela 7.10 apresenta medidas de confiabilidade, segurança e

disponibilidade, obtidas para diversos valores dos fatores de cobertura e das taxas de falha.

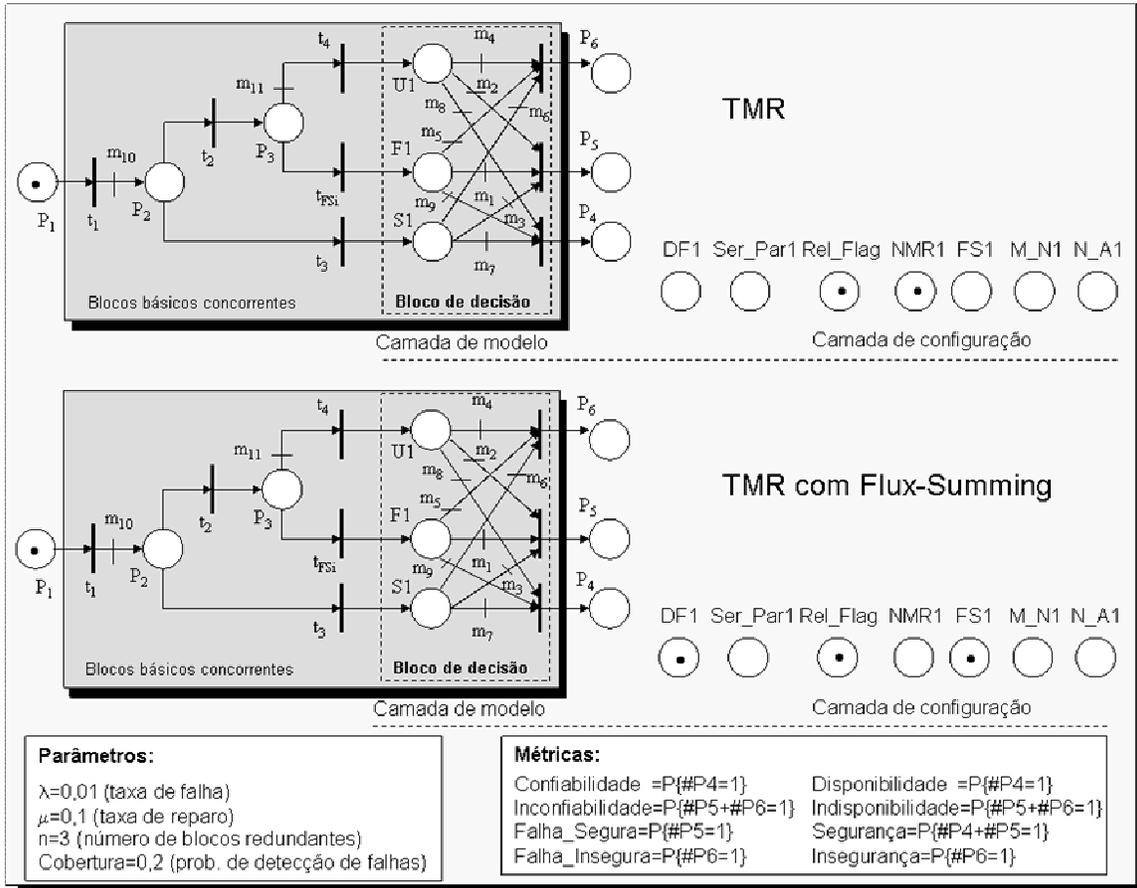


Figura 7.19 Modelos MDP das configurações TMR e TMR com *Flux-Summing*

Tabela 7.10 Estimativas de dependabilidade da configuração TTMR

$\lambda=0.01$ falhas/hora $\mu=0.1$ reparos/hora CF =Fator de Cobertura			$\lambda=0.001$ falhas/hora $\mu=0.1$ reparos/hora CF =Fator de Cobertura			$\lambda=0.0001$ falhas/hora $\mu=0.1$ reparos/hora CF =Fator de Cobertura		
Disponibil.: 0.95181225			Disponibil.: 0.99941288			Disponibil.: 0.99999401		
CF	Confiabil.	Segurança	Confiabil.	Segurança	Confiabil.	Segurança	Confiabil.	Segurança
0.0	0.98722197	0.99486906	0.99986568	0.99994627	0.99999865	0.99999946	0.99999865	0.99999946
0.1	0.98729748	0.99486898	0.99986649	0.99994627	0.99999866	0.99999946	0.99999866	0.99999946
0.2	0.98752403	0.99486844	0.99986890	0.99994627	0.99999868	0.99999946	0.99999868	0.99999946
0.3	0.98790169	0.99486698	0.99987293	0.99994627	0.99999872	0.99999946	0.99999872	0.99999946
0.4	0.98843058	0.99486414	0.99987856	0.99994626	0.99999878	0.99999946	0.99999878	0.99999946
0.5	0.98911086	0.99485945	0.99988581	0.99994626	0.99999885	0.99999946	0.99999885	0.99999946
0.6	0.98994274	0.99485244	0.99989466	0.99994625	0.99999894	0.99999946	0.99999894	0.99999946
0.7	0.99092647	0.99484267	0.99990513	0.99994624	0.99999905	0.99999946	0.99999905	0.99999946
0.8	0.99206235	0.99482965	0.99991721	0.99994623	0.99999917	0.99999946	0.99999917	0.99999946
0.9	0.99335074	0.99481292	0.99993089	0.99994621	0.99999931	0.99999946	0.99999931	0.99999946
1.0	0.99479200	0.99479201	0.99994619	0.99994619	0.99999946	0.99999946	0.99999946	0.99999946

No cálculo das medidas de dependabilidade da Tabela 7.10 foram considerados os mesmos valores de taxas de falha e de reparo para todos os blocos componentes de uma configuração TMR ou TMR com *flux-summing*, para diversos fatores de cobertura. Através dos modelos MDP, medidas de dependabilidade podem ser calculadas para diferentes valores de taxas de falha e de reparo dos blocos, e diferentes fatores de cobertura, sem que haja necessidade de alteração do modelo, apenas dos parâmetros. Isto fornece aos modelos MDP grande flexibilidade e permite a reusabilidade dos modelos. Na Figura 7.20 é mostrada uma imagem do modelo MDP, gerada pela ferramenta de modelagem, para obtenção das medidas da configuração TTMR. Esta figura mostra os cinco blocos da Figura 7.18, representados pelo modelo da Figura 7.19, cada qual com sua própria camada de modelo e de configuração, além dos parâmetros estruturais. O bloco dos atuadores mecânicos não é mostrado nesta figura, uma vez que sua confiabilidade é constante e igual a 0,97.

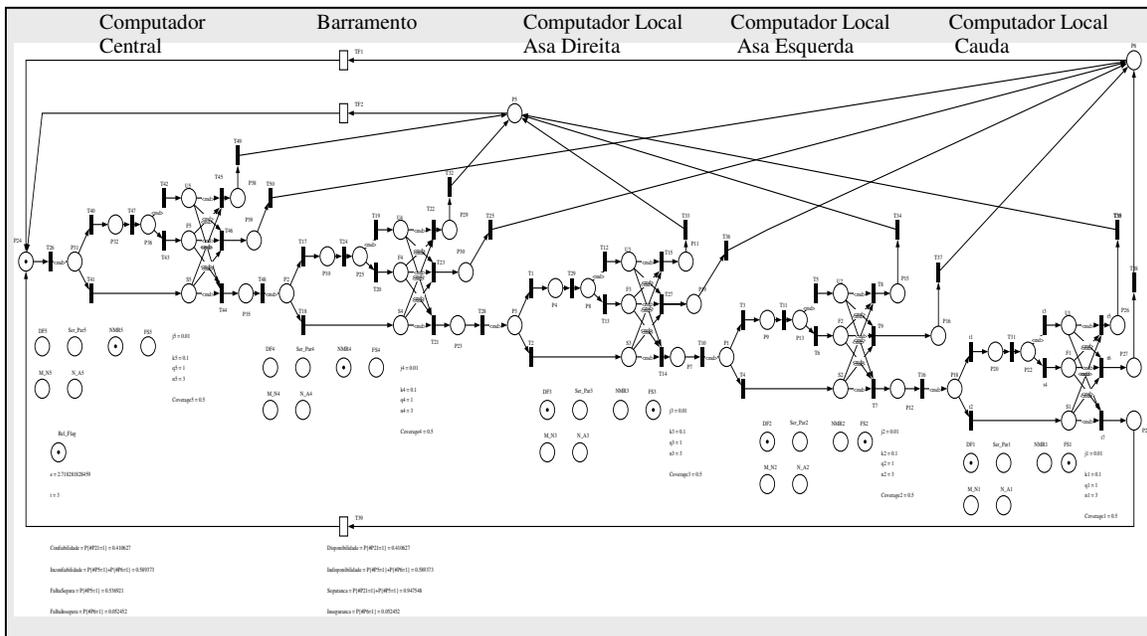


Figura 7.20 Modelo MDP da configuração TTMR do sistema de controle de voo

No modelo MDP, as transições são imediatas, e os pesos podem ser definidos por meio de expressões analíticas ou numéricas. Para possibilitar medidas analíticas de estado permanente ou transiente, são colocadas transições temporizadas, denominadas transições de *feedback*. A análise de estados transientes é feita por meio de um conjunto de análises de estados permanentes para tempos discretos em intervalos bem definidos. Esta representação por meio de rede de Petri do diagrama de blocos de um sistema dependável, permite a obtenção de medidas de confiabilidade, disponibilidade e segurança, através da análise de estado permanente ou transiente, reduzindo significativamente o número de estados.

- **Técnica TDTMR:** na solução de projeto Triplo-Duplex TMR (TDTMR), a arquitetura contém três canais independentes conforme mostrado na Figura 7.21. Cada canal consiste de um conjunto de sensores e computadores com configuração *duplex* com

comparação. A solução triplo-*duplex* é uma combinação dos mecanismos de tolerância a falhas de duplicação com comparação (*duplex* com comparação) e TMR, conforme descrito no capítulo 3. A configuração *duplex* com comparação empregada como técnica de detecção de falha é utilizada para o computador central, para o computador local da asa esquerda, para o computador local da asa direita e para o computador local da cauda da aeronave. Os computadores locais estão interligados ao computador central numa estrutura ponto a ponto. Nesta solução, a única possibilidade de mascaramento de falha é patrocinada pela técnica de *flux-summing*. Uma falha em qualquer um dos computadores interconectados, torna o respectivo canal inoperante. Esta solução pode suportar mais de uma falha desde que estas sejam detectadas e o canal correspondente seja removido do processo de *flux-summing*. Considera-se que a taxa de falha de um computador numa configuração *duplex* com comparação é aproximadamente o dobro daquela de um computador único, isto é, $\lambda_{duplex} = 2\lambda$.

A arquitetura TDTMR é composta por 3 canais numa configuração TMR com *flux-summing*, sendo cada canal composto por um computador central e três computadores locais em série, cada qual numa configuração *duplex* com comparação. Portanto a taxa de falha de cada canal é de aproximadamente 8λ . O modelo TDTMR utilizado na ferramenta de modelagem é semelhante ao modelo TMR com *flux-summing* mostrado na Figura 7.19, porém com a taxa de falha $\lambda_{TDTMR} = 8\lambda$, onde λ é a taxa de falha de cada um dos computadores, por hipótese, igual para todos. As medidas obtidas são mostradas na Tabela 7.11.

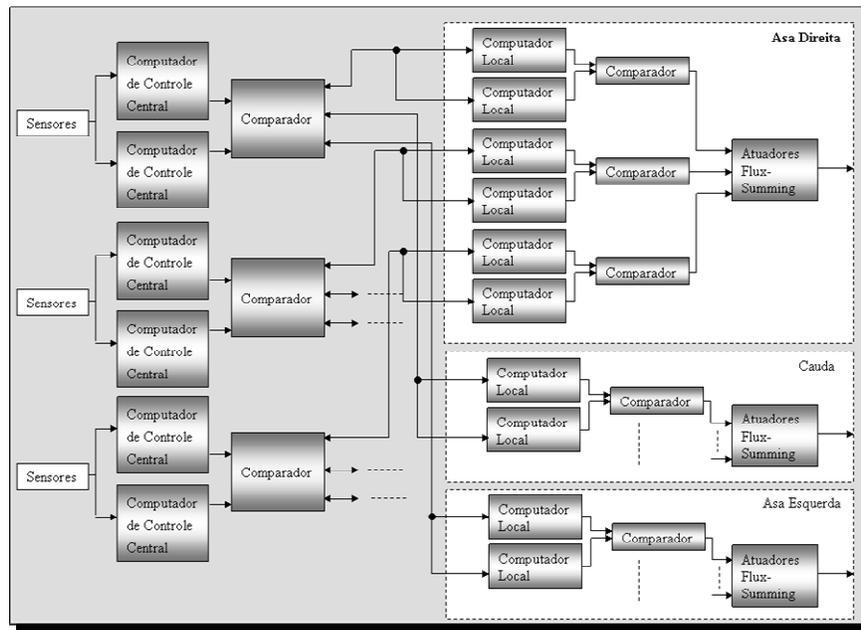


Figura 7.21 Arquitetura candidata TDTMR

- **técnica 5MR:** nesta solução de projeto, a arquitetura contém 5 canais completamente independentes, numa configuração *flux-summing*, conforme descrito na Figura 7.22. Os computadores centrais estão interconectados aos computadores locais numa topologia ponto a ponto. Até 2 falhas são suportadas nesta solução sem necessidade

de detecção de falha. Mais do que 2 falhas podem ser suportadas, desde que detectadas e retiradas da configuração *flux-summing*. Uma terceira falha não será tolerada a menos que pelo menos duas das três falhas sejam detectadas e os respectivos canais removidos da configuração *flux-summing*. Se houver detecção e reconfiguração, até quatro falhas podem ser toleradas, aumentando deste modo, significativamente a confiabilidade e a segurança do sistema.

Tabela 7.11 Estimativas de dependabilidade da configuração TDTMR

$\lambda=8*(0.01\text{falhas/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura			$\lambda=8*(0.001\text{falhas/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura		$\lambda=8*(0.0001\text{falhas/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura	
Disponibil.: 0.91220785			Disponibil.: 0.99959353		Disponibil.: 0.99999950	
CF	Confiabil.	Segurança	Confiabil.	Segurança	Confiabil.	Segurança
0.0	0.88284565	0.88284565	0.99833957	0.99833957	0.99998279	0.99998279
0.1	0.88392005	0.88392977	0.99835604	0.99835605	0.99998296	0.99998296
0.2	0.88714325	0.88722097	0.99840545	0.99840556	0.99998348	0.99998348
0.3	0.89251525	0.89277754	0.99848781	0.99848817	0.99998434	0.99998434
0.4	0.90003605	0.90065777	0.99860310	0.99860396	0.99998554	0.99998554
0.5	0.90970565	0.91091995	0.99875134	0.99875301	0.99998709	0.99998709
0.6	0.92152406	0.92362235	0.99893252	0.99893540	0.99998898	0.99998898
0.7	0.93549126	0.93882327	0.99914665	0.99915122	0.99999122	0.99999122
0.8	0.95160726	0.95658100	0.99939371	0.99940054	0.99999380	0.99999380
0.9	0.96987206	0.97695381	0.99967372	0.99968344	0.99999672	0.99999673
1.0	0.99028545	0.99999976	0.99998666	1.00000000	0.99999999	1.00000000

A arquitetura 5MR é composta por 5 canais numa configuração 5MR com *flux-summing*, sendo cada canal composto por um computador central e três computadores locais em série, acarretando numa taxa de falha de cada canal de 4λ .

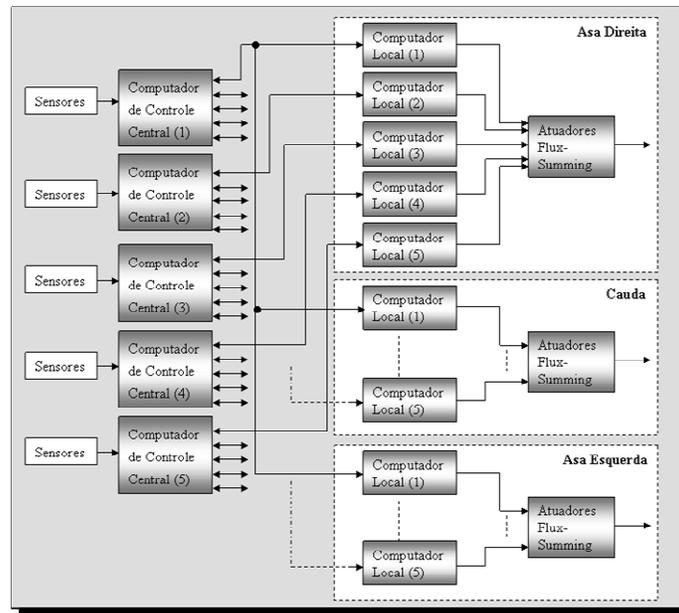


Figura 7.22 Arquitetura candidata 5MR

O modelo 5MR com *flux-summing*, utilizado na ferramenta de modelagem, correspondente ao modelo TMR com *flux-summing* mostrado na Figura 7.19, exceto que agora o número de réplicas concorrentes, n , será 5 ao invés de 3. As estimativas obtidas são mostradas na Tabela 7.12.

Tabela 7.12 Estimativas de dependabilidade da configuração 5MR

$\lambda=4*(0.01\text{ falhas/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura			$\lambda=4*(0.001\text{ falhas/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura		$\lambda=4*(0.0001\text{ falha/hora})$ $\mu=0.1\text{ reparos/hora}$ CF =Fator de Cobertura	
Disponibil.: 0.99809594			Disponibil.: 0.99999992		Disponibil.: 1.0000000	
CF	Confiabil.	Segurança	Confiabil.	Segurança	Confiabil.	Segurança
0.0	0.98788220	0.98788220	0.9999833	0.99998333	0.999999	0.9999999
0.1	0.98820075	0.98820075	0.9999838	0.99998379	0.999999	0.9999999
0.2	0.98906628	0.98906628	0.9999851	0.99998505	0.999999	0.9999999
0.3	0.99034490	0.99034495	0.9999869	0.99998691	0.999999	0.9999999
0.4	0.99190449	0.99190468	0.9999892	0.99998917	0.999999	0.9999999
0.5	0.99361463	0.99361521	0.9999916	0.99999162	0.999999	0.9999999
0.6	0.99534667	0.99534810	0.9999941	0.99999408	0.999999	0.9999999
0.7	0.99697368	0.99697679	0.9999963	0.99999634	1.000000	1.0000000
0.8	0.99837050	0.99837656	0.9999982	0.99999822	1.000000	1.0000000
0.9	0.99941367	0.99942459	0.9999995	0.99999950	1.000000	1.0000000
1.0	0.99998151	1.00000000	1.0000000	1.00000000	1.000000	1.0000000

Conforme pode ser observado na Tabela 7.12, há uma melhoria significativa da confiabilidade, disponibilidade e segurança, porém a um custo também maior. Este custo será reflexo da maior quantidade de equipamentos, da maior dimensão do sistema, do maior peso e do maior consumo de energia, dentre outros incrementos. A escolha da arquitetura mais adequada ao sistema será uma solução de compromisso entre os requisitos de dependabilidade e os custos envolvidos. Não é objetivo deste estudo de caso inferir possíveis análises comparativas das estimativas obtidas como foi feito originalmente. Objetiva-se apenas a validação dos valores obtidos e não a solução de projeto mais adequada, o que poderá ser feito em trabalhos futuros.

7.4 Estudo de Caso IV: Utilização de Mecanismo Tolerante a Falhas na Implementação de um Sistema de Ordenação Composto por Hw e Sw

Este estudo de caso aborda um sistema de ordenação composto por componentes de hardware e de software utilizando técnicas de tolerância a falhas do tipo TMR (redundância modular tripla).

Os passos iniciais da metodologia correspondem a especificação do sistema de ordenação por meio do mecanismo de tolerância a falhas TMR, constituído por quatro blocos, três dos quais compostos por réplicas semelhantes, porém distintas de hardware, e versões distintas de software, e um quarto bloco formado por um mecanismo de votação em software executado sobre um dispositivo de hardware. As diferentes versões de

software do sistema correspondem aos algoritmos de ordenação *insert*, *bubble* e *gnome*. As réplicas de hardware são semelhantes e correspondem ao microcontrolador 8051 da Intel, implementadas por meio do simulador Proteus, conforme mostrado na Figura 7.23.

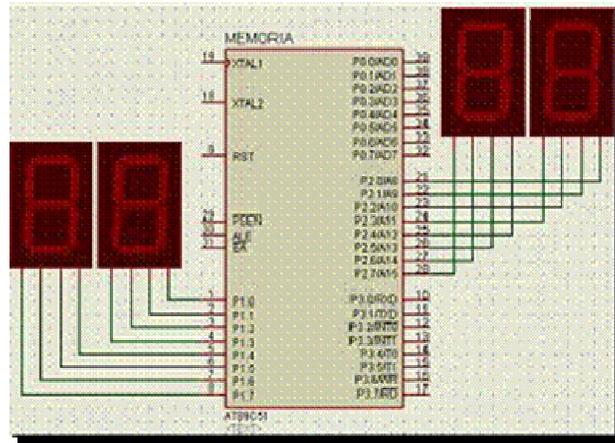


Figura 7.23 Microcontrolador 8051

As probabilidades de falha, correspondentes aos microcontroladores 8051 e aos softwares de ordenação em cada bloco, são determinadas por meio de geradores de números aleatórios distintos. Neste estudo de caso, a probabilidade de falha de cada um dos dispositivos microcontroladores 8051 é de 0,01, enquanto as probabilidades de falha dos diferentes algoritmos de ordenação é de 0,05. A representação em blocos do sistema de ordenação por meio do mecanismo de tolerância a falhas TMR é mostrada na Figura 7.24. Na Figura 7.25 a simulação do sistema de ordenação, configurado de acordo com o mecanismo de tolerância a falhas TMR, considerando-se as diversas versões de software e réplicas do microcontrolador 8051, é apresentada, conforme utilização no simulador Proteus.

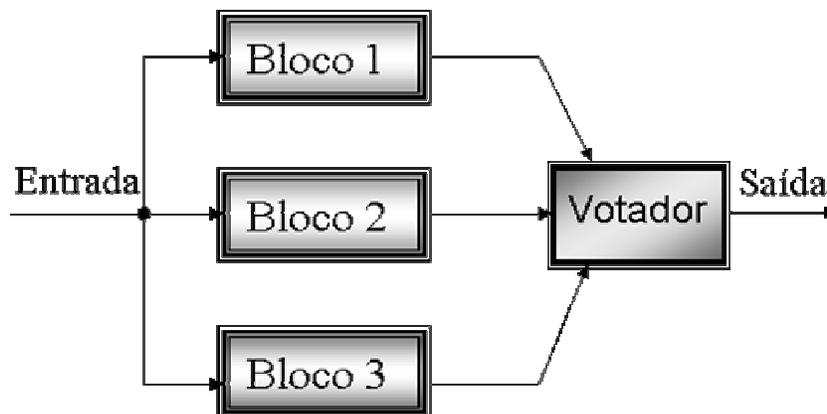


Figura 7.24 Sistema de ordenação com mecanismo TMR

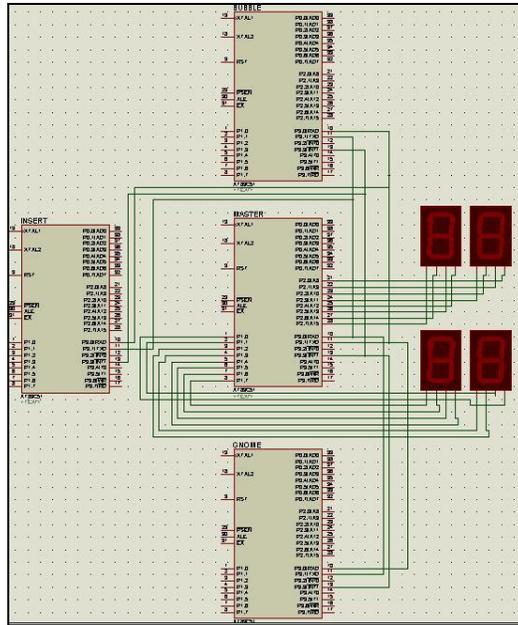


Figura 7.25 Simulação do sistema de ordenação por meio da técnica TMR

O diagrama de blocos intermediário, que define a composição dos diversos blocos na formação do mecanismo TMR, é mostrado conforme a Figura 7.26.

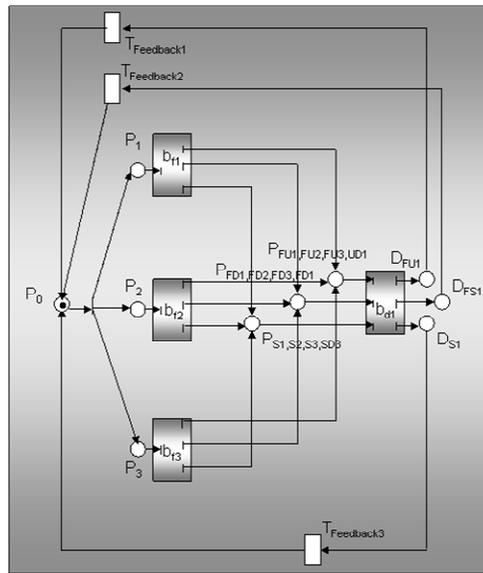


Figura 7.26 Diagrama de blocos intermediário

Os blocos básicos funcionais b_{f1} , b_{f2} e b_{f3} representam as versões dos softwares de ordenação executadas sobre réplicas do microcontrolador 8051, enquanto o bloco b_{d1} representa o mecanismo de votação executado no microcontrolador 8051.

O modelo de cada uma das réplicas do sistema de ordenação, constituído por componentes de hardware e de software, é configurado conforme a Figura 7.27.

O componente de software, correspondente a uma das rotinas de ordenação, é único, assim como o componente de hardware que o suporta.

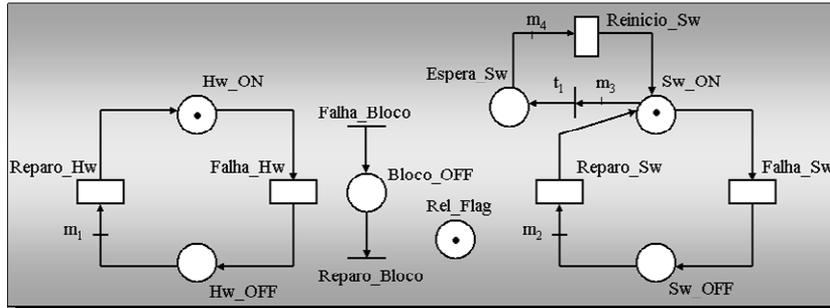


Figura 7.27 Componente de Sw único em componente de Hw único

Na Figura 7.27, as funções correspondentes a falha, reparo e reinício apresentam distribuições exponenciais. Neste modelo apenas as métricas de confiabilidade e disponibilidade são avaliadas uma vez que o sistema não leva em consideração a cobertura de falhas. Considera-se que todas as ocorrências de falhas são passíveis de detecção. Uma vez avaliado o modelo da Figura 7.27, para cada um dos blocos, são obtidos os valores de confiabilidade a serem utilizados no modelo MDP. A configuração do bloco, composto por hardware e software, corresponde ao modo serial na avaliação de confiabilidade, uma vez que a falha de qualquer um dos componentes produzirá uma falha no bloco correspondente. Portanto, a confiabilidade do sistema de ordenação é de 0.98980054, considerando-se o bloco votador livre de falhas, a confiabilidade de cada bloco de 0.9405 e um tempo de missão de 1u.t. (u.t. significa uma unidade de tempo).

O próximo passo consiste na construção do modelo MDP capaz de representar os eventos de falha de cada um dos blocos do sistema de ordenação e o mecanismo de votação. A confiabilidade de cada bloco do sistema de ordenação, obtida através do modelo da Figura 7.27, é utilizada no modelo MDP mostrado na Figura 7.28. O mecanismo de decisão no modelo MDP é representado pelo bloco de decisão, conforme observado na Figura 7.28.

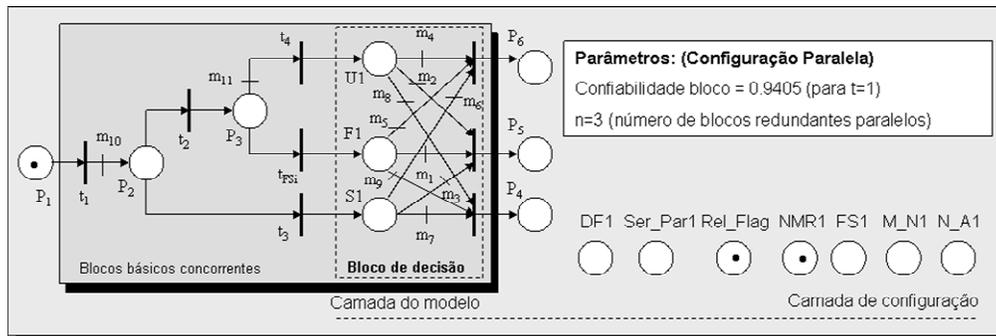


Figura 7.28 Modelo MDP para sistema de ordenação com mecanismo TMR

Os resultados da confiabilidade e da inconfiabilidade dos blocos, compostos pelos microcontroladores 8051 e diferentes versões do software de ordenação, são exibidos na Tabela 7.13. Os resultados obtidos através do modelo da Figura 7.28 são comparados aos

resultados obtidos por meio de simulação, de acordo com o circuito da Figura 7.25, conforme mostrado na Tabela 7.14.

Tabela 7.13 Confiabilidade dos blocos do sistema de ordenação

Componente	Inconfiabilidade	Confiabilidade
Hw Bloco 1	0,0100796568627	0,9899203431373
Sw Insert	0,0505091145430	0,949490885457
Hw Bloco 2	0,0101715686274	0,9898284313726
Sw Bubble	0,0498173826916	0,9501826173084
Hw Bloco 3	0,0096813725490	0,9903186274510
Sw GNome	0,0495297611682	0,9504702388318

Tabela 7.14 Comparação de resultados: modelo x simulador

Mecanismo TMR	Inconfiabilidade	Confiabilidade
Simulação Proteus	0,01017595997	0,98982404003
Modelo Hw/Sw	0,01019946025	0,98980053975

Os resultados obtidos pelo modelo de hardware e software tolerante a falhas TMR e pelo simulador apresentam resultados bastante próximos o que valida o modelo em questão e a metodologia. Para a modelagem de dependabilidade de sistemas mais complexos, compostos por diferentes configurações de hardware e software, podem ser utilizados os modelos apresentados no Capítulo 6.

Considerações Finais

A metodologia original, por meio das cadeias de Markov e das expressões analíticas, obteve valores de confiabilidades para as quatro arquiteturas aqui mostradas, considerando-se o mesmo valor da taxa de falha para todos os componentes da arquitetura. A metodologia proposta permite não apenas os cálculos de confiabilidade para diversos valores do fator de cobertura, como também possibilita o cálculo de disponibilidade e segurança. Além disso, permite que a taxa de falha de qualquer componente possa assumir qualquer valor, uma vez que o modelo é parametrizado. As medidas de confiabilidade, disponibilidade e segurança, para qualquer uma das arquiteturas candidatas, são obtidas por meio da ferramenta de modelagem em um tempo computacional bastante reduzido. Nestes estudos de caso, estimativas de confiabilidade, disponibilidade e segurança foram obtidas por meio da metodologia e dos modelos apresentados nos Capítulos 4, 5 e 6 desta Tese. As estimativas dos modelos analisados são validadas por meio dos valores originalmente obtidos, sendo apresentadas em forma de tabelas para diferentes valores das taxas de falha dos componentes da arquitetura candidata e diferentes valores dos fatores de cobertura. Os resultados apresentados não apenas corroboram aqueles obtidos originalmente para confiabilidade, como também estendem as tabelas acrescentando novas medidas relativas aos atributos de disponibilidade e segurança.

Capítulo 8

Conclusão e Trabalhos Futuros

8.1 Conclusão

Apesar dos esforços feitos ao longo dos anos, a dependabilidade dos sistemas tem estado abaixo do que se esperava e em algumas situações até pior. Explica-se, em parte esta afirmação, com base na complexidade alcançada pelos sistemas computacionais e sua integração com as redes de comunicação e serviço, envolvendo desde pequenos sistemas embarcados a grandes redes de arquiteturas abertas. Deste modo, para que se possa modelar e avaliar a dependabilidade dos sistemas envolvidos nesta contínua e complexa evolução tecnológica, levando-se em conta combinações de atributos e aspectos de vulnerabilidade dos sistemas contra intrusões, não apenas os métodos atuais devem ser reconsiderados, estudados, explorados e avaliados, como também novos métodos de modelagem e simulação devem ser desenvolvidos, para fazer frente aos cenários de dependabilidade dos sistemas atuais e futuros.

Os objetivos principais desta Tese podem ser classificados como: o desenvolvimento de uma metodologia de modelagem e avaliação de sistemas dependáveis, e a definição de um conjunto (biblioteca) de modelos.

A biblioteca de modelos foi construída baseada em redes EDSPN, e em modelos de redes de Petri EDSPN estruturados na forma produto, denominados MDP. Os modelos MDP estão associados aos diferentes tipos de blocos presentes em um diagrama de blocos, porém com mecanismos de tolerância a falhas incorporados;

O desenvolvimento de uma metodologia híbrida e modular possibilitou a modelagem, o refinamento e a avaliação de dependabilidade dos sistemas, de um modo flexível e expansível, buscando ocultar a complexidade matemática envolvida e reduzir características indesejáveis, como a possibilidade de explosão de estados e *stiffness*.

Alguns dos objetivos específicos alcançados por esta Tese foram: o refinamento de modelos, mecanismos de composição e modelagem de estratégias de reparo.

A metodologia apresentada para avaliação e modelagem de sistemas dependáveis, de forma hierárquica, fundamentada em diagrama de blocos e redes de Petri, está baseada, essencialmente, em três conceitos fundamentais: modularidade, hierarquia e parametrização.

A modularização teve como objetivo facilitar a análise estrutural do sistema por meio de seu particionamento em um conjunto de blocos interconectados. O

particionamento foi executado considerando-se a criticidade de cada bloco com relação aos requisitos de dependabilidade e a configuração do sistema.

A hierarquização teve como objetivo facilitar a análise do sistema, pela evolução do nível de detalhes fornecidos pelos modelos, representativos do sistema em cada nível hierárquico.

- No nível hierárquico inicial o sistema é representado por um diagrama de blocos pela aplicação do conceito de modularidade;
- No segundo nível hierárquico, ao diagrama de blocos original foram incorporados novos blocos, com base nos mecanismos de tolerância a falha atribuídos aos blocos considerados críticos, e com base também na geração de blocos especiais, denominados blocos de decisão. Os blocos de decisão foram introduzidos, com base em uma estrutura cruzada e expressões lógicas condicionais, para solucionar problemas de configurações do tipo não-serial/não-paralela no diagrama de sistema dependável (EDBD) e possibilitar a obtenção de estimativas de dependabilidade;
- No terceiro nível hierárquico foram definidas regras de redução do modelo. Estas regras de redução quando aplicadas às redes de Petri, possibilitam a redução da complexidade, dado a redução do grafo de alcançabilidade, contudo preservando propriedades qualitativas do modelo tais como *liveness*, limitação e *safeness* ;
- A partir do quarto nível, cada bloco individualmente, ou um conjunto de blocos com suas dependências, são analisados isoladamente. Adotou-se o modelo EDSPN, dado a possibilidade de se ter os arcos anotados com variáveis que representam os tempos médios de retardo e multiplicidade dos pesos, em função da marcação. A adoção deste modelo proporcionou um maior poder de representação, pela possibilidade de refinamento de lugares, eventos, componentes e estratégias de reparo. Neste nível, foram obtidas expressões numéricas ou analíticas de dependabilidade, por meio de avaliações de estado permanente ou transiente, aplicadas aos modelos EDSPN;
- No quinto nível hierárquico foram definidos os modelos, denominados MDP, para os blocos do diagrama EDBD. Estes modelos utilizam as expressões analíticas ou numéricas obtidas dos modelos EDSPN e os coloca nos pesos das transições imediatas por meio de expressões lógicas condicionais. As configurações dos modelos MDP para geração do diagrama de dependabilidade do sistema, são baseados no diagrama de blocos intermediários, obtidas no terceiro nível hierárquico. De posse dos detalhes gerados nos terceiro e quarto níveis, o diagrama de dependabilidade do sistema foi gerado. A partir da obtenção do diagrama dependável, estimativas de dependabilidade do sistema puderam ser calculadas de modo flexível, pela utilização de parâmetros, e com redução do espaço de estados e *stiffness*.

A biblioteca de modelos gerada permite a um modelador construir novos modelos com pequenas modificações. Parte destas modificações ocorrem nas regras lógicas condicionais, as quais são expressões do tipo IF/THEN/ELSE, cuja sintaxe está definida no ANEXO-B. Por meio de reutilização, novos modelos podem ser construídos, o que permite não apenas agilizar as atividades da modelagem, como também reduzir as possibilidades de erros de representação. Caso se deseje utilizar modelos quantitativos já existentes, e representá-los diretamente aos modelos MDP, é suficiente atribuir suas expressões analíticas ou numéricas aos pesos das transições imediatas e conflitantes do modelo. Deste modo tendo-se expressões analíticas ou expressões numéricas de dependabilidade, correspondentes a cada bloco do digrama EDBD, pode-se calcular os valores dos atributos de dependabilidade do sistema;

Dentre as contribuições oferecidas por esta Tese, podem ser destacadas as seguintes:

- o desenvolvimento de uma metodologia que possibilitou a modelagem e a análise de sistemas dependáveis. Devido a adoção de variáveis parametrizadas, as quais são utilizadas em várias configurações de um mesmo modelo, ao invés de valores fixos, esta metodologia pôde modelar diferentes configurações do sistema. Nos estudos de caso, a metodologia na avaliação foi aplicada a sistemas em operação ou ainda na fase de projeto. O processo de análise foi realizado com geração do espaço de estados de forma controlada e com controle de *stiffness*. Por apresentar um conjunto de parâmetros de configuração e estrutural, esta metodologia é passível de automação por meio de ferramentas de níveis mais altos, facilitando a utilização por pessoal menos especializado;
- a criação de uma biblioteca de modelos, os quais dependendo dos valores das taxas, do número de blocos, e dos parâmetros de configuração, podem assumir diferentes representações. As biblioteca de modelos foram desenvolvidas para os modelos correspondentes aos níveis hierárquicos 4 e 5. Os modelos desenvolvidos foram função das várias formas de replicação ou redundância, dos mecanismos de tolerância a falhas: a) replicação passiva, com réplica inativa livre de falhas; b) replicação passiva, com réplica inativa sujeita a falhas; c) replicação semi-ativa; d) replicação ativa NMR; e) replicação ativa generalizada, utilizadas na modelagem de um grande número de sistemas.
- o desenvolvimento de modelos únicos para representação de mecanismo tolerante a falhas, com diferentes níveis de redundância e diferentes taxas de falha e reparo, ou ainda, o desenvolvimento de modelos únicos para representação de diferentes mecanismos tolerante a falhas, também é uma contribuição desta Tese;
- uma contribuição essencial em todo este processo de modelagem, foi a construção de regras de decisão baseadas nos mecanismos de tolerância a falhas e nas configurações dos diversos blocos que compõem o modelo. Estas regras de decisão, colocadas nos modelos MDP, são utilizadas para criar as condições de sucesso, falha segura (detectável) ou falha insegura (não-detectável) quando na

avaliação de um bloco, de um conjunto de blocos, ou do sistema como um todo. Estas regras são expressões lógicas condicionais criadas em função das marcações, incluindo os *flags* de configuração, e os parâmetros estruturais. Por meio dessas regras lógicas o diagrama de dependabilidade do sistema pode ser depurado por partes, e pode ser reconfigurado, considerando-se as diversas formas de redundância, até a obtenção de valores válidos;

- uma outra contribuição foi a possibilidade de avaliação de grafos transientes, através de análise de estado permanente do sistema, considerando-se intervalos de tempo regulares, denominados pontos de verificação, para avaliações num intervalo de tempo de 0 a t ;
- a possibilidade de resolver através de mecanismos gráficos, diagramas que não apresentavam configurações exclusivamente serial e paralela, também é uma contribuição deste trabalho. A solução para este tipo de problema foi obtida pela criação dos blocos de decisão;
- o desenvolvimento de mecanismos para composição de modelos, que possibilitou as formas de interações entre os blocos, é outra contribuição. Estas interações entre blocos intermediários, foram geradas pela aplicação das regras de redução, que resultou num diagrama de blocos intermediários. Deste modo podem-se substituir as transições imediatas do bloco intermediário por uma rede de Petri associada e gerar o diagrama de dependabilidade do sistema;
- definição de modelos de reparo que representem estratégias de manutenção, também é uma contribuição do presente trabalho;

Devido à dificuldade de modelagem de blocos com um mesmo comportamento, porém com diferentes taxas de falha e reparo, em um modelo único EDSPN, foram criados os modelos MDP. No modelo MDP blocos com mesmo comportamento, porém com diferentes taxas podem ser representados por um modelo único. Também mecanismos tolerantes a falhas similares, com diferentes números de componentes redundantes, podem ser representados por um único modelo MDP. Para isso foram criadas duas camadas neste modelo: a camada de modelo e a camada de configuração. Na camada de modelo as expressões analíticas ou numéricas, obtidas do modelo EDSPN, são inseridas. Na camada de configuração foram definidos *flags*, cujas ativações ou desativações atribuem aos blocos diferentes configurações e mecanismos de tolerância a falhas. Estas camadas conferem aos modelos flexibilidade e possibilidade de reconfiguração, pela alteração das regras lógicas condicionais nas representações dos atributos e no mecanismo de decisão.

A metodologia descrita nesta Tese apresentou algumas características incomuns da maioria das metodologias de avaliação de dependabilidade por meio de redes de Petri. Nesta metodologia foram utilizados parâmetros ao invés de valores fixos para as taxas de falha, reparo e percepção, ou ainda para os tempos médios até a ocorrência de falha MTTF, tempos médios para reparo MTTR, e tempos médios de percepção MTEP, além

do fator de cobertura, essencial na definição do percentual das falhas detectadas e não-detectadas. Estes parâmetros podem assumir valores diversos, de acordo com a classe de representação numérica a que pertence. Por exemplo, as variáveis correspondentes às taxas de falha (j_i), reparo (k_i) e percepção (k_j) ou os respectivos tempos médios, podem assumir qualquer valor no campo dos números reais positivos, enquanto as variáveis correspondentes aos fatores de cobertura (C_i), podem assumir qualquer valor real entre 0 e 1, inclusive, os quais representam probabilidades de detecção de falhas. Os parâmetros correspondentes ao número de réplicas de um determinado mecanismo de tolerância a falhas podem adotar qualquer valor inteiro maior ou igual a 1, enquanto a variável q , que representa o número mínimo de componentes a partir do qual a configuração paralela m/n , gera uma falha, pode assumir qualquer valor inteiro maior ou igual a 1. Neste trabalho também foram definidos parâmetros de configuração, ou *flags*, os quais representam o tipo de mecanismo de tolerância a falhas a ser utilizado no modelo, como por exemplo, *N-modular redundancy* (NMR), *flux-summing* (FS), *coldstandby* (CS), *warmstandby* (WS) e *hotstandby* (HS). Os *flags* também representam as diversas formas de configuração estrutural do sistema, o tipo de avaliação a ser executada e o modo de avaliação gerado.

Os exemplos utilizados nos estudos de caso descreveram sistemas com características distintas, demonstrando as possibilidades de modelagem da metodologia. Um dos estudos de caso, voltado aos sistemas de controle de vôo de aeronaves, enfatizou os altos requisitos de confiabilidade deste tipo de sistema e estendeu o modelo original, através de estimativas de disponibilidade e segurança. Neste estudo de caso, o processo de avaliação analítica iniciado ainda na fase de projeto, analisou decisões de projeto entre soluções alternativas, por meio da utilização da metodologia e dos modelos propostos. O estudo de caso relativo ao circuito de disparo do motor de um foguete lançador de mísseis, demonstrou a possibilidade de transformação de modelos, originalmente escritos utilizando técnicas combinatoriais, através do modelo de árvore de falhas, para a metodologia em questão. Ao requisito de confiabilidade do modelo original, foram acrescentadas as estimativas relativas aos atributos de disponibilidade e de segurança, permitindo que diversas análises de sensibilidade fossem realizadas com os modos e os eventos de falha do sistema. A avaliação dos sistemas de operação contínua, os quais possuem elevados requisitos de disponibilidade, foi representada pelo estudo de caso envolvendo o sistema de telecomunicações em suporte a um sistema de transmissão de energia elétrica. Ao estudo de caso original envolvendo a avaliação dos requisitos de disponibilidade, formado por dois trechos da rede de telecomunicações, foram acrescentados requisitos de confiabilidade e segurança. Para todos os três estudos de caso descritos nesta Tese, foram realizadas avaliações de confiabilidade, disponibilidade e segurança, através de análise de estado permanente e transiente, envolvendo expressões analíticas e numéricas.

Os modelos e a metodologia apresentados seguem as tendências atuais na direção dos ambientes de dependabilidade [129]. A solução apresentada é uma solução hierárquica, a qual lida com características intrínsecas dos modelos Markovianos, como a explosão de estados e *stiffness*. Além disso, a capacidade de selecionar e montar modelos de componentes por meio de diversas combinações, em um modelo geral (diagrama),

segue as metas de: a) construção de uma biblioteca de modelos para diferentes tipos de componentes; b) geração automática de um modelo apropriado; c) composição automática de modelos em um dado nível hierárquico, por meio de regras de composição, as quais são dependentes da aplicação, para a definição do modelo geral.

8.2 Trabalhos Futuros

Na metodologia apresentada foram avaliadas estimativas de confiabilidade, disponibilidade e segurança. A análise de uma rede de comunicação considerando requisitos de desempenho dissociados dos requisitos de disponibilidade ou confiabilidade tende a ser otimista. Avaliação de atributos compostos tende a fornecer resultados mais realistas a respeito dos sistemas. Deste modo, um dos trabalhos futuros é a extensão dos atributos do sistema a serem avaliados, pela introdução de requisitos compostos, tais como performabilidade, o qual avalia o desempenho em função da degradação do sistema;

A complexidade dos novos dispositivos computacionais fixos ou móveis, e das redes de comunicação e serviço integradas, apresentam além dos componentes de hardware e de software, um novo e imprevisível componente: o ser humano (usuários, operadores, reparadores e estranhos). Portanto, para lidar com este novo componente, a metodologia apresentada para avaliação da dependabilidade dos sistemas deve ser estendida, para possibilitar a avaliação de estimativas de segurança dos sistemas contra intrusão.

Um outro trabalho de pesquisa a ser desenvolvido objetiva a análise de sistemas de tempo real com a introdução no modelo de restrições críticas de tempo. A composição dos modelos de hardware e software, aliada as restrições de tempo, pode permitir a análise dos sistemas embarcados e possibilitar a avaliação de uma função de custo composta por estimativas de confiabilidade, disponibilidade, segurança, desempenho e potência.

Uma outra linha de pesquisa a ser desenvolvida envolve a implementação de ferramentas de modelagem de alto nível com o objetivo de tornar a interface mais amigável para avaliação da dependabilidade dos sistemas, através da passagem de parâmetros. Desse modo, os conceitos serão melhor compreendidos, diminuindo o *gap* entre o que se conhece e o que se faz neste campo da pesquisa.

Referências Bibliográficas

- [1] Agerwala, T., Choed-Amphai, Y., “A synthesis rule for concurrent systems,” In *Proceedings of the 15th Design Automation Conference*, Las Vegas, pages 305-311, June 1978.
- [2] Almeida, C.B., “Construction et Affinement de modèles de sûreté de fonctionnement – Application aux systèmes de controle-commande,” Thèse de Doctorat, LASS-CNRS, Juin 2002.
- [3] Almeida, C.B., Kanoun, K., “Dependability modelling of instrumentation and control systems: a comparison of competing architectures”, *LAAS-CNRS Report 02204*, 2003.
- [4] Almeida, C.B., Kanoun, K., “Construction and stepwise refinement of dependability models”, In *Performance Evaluation*, Vol. 56, pages 277-306, 2004.
- [5] Ammar, H., Huang, Y., Liu, R. “Hierarchical Models for Systems Reliability, Maintainability, and Availability”, In *IEEE Transactions on Circuits and Systems*, Vol. CAS-34, NO. 6, pages 629-638, June 1987.
- [6] Anderson, T. and Lee, P. A. “*Fault Tolerance: Principles and Practice*”, Springer-Verlag, Wien - New York, 1990.
- [7] Anzai, K., Adachi, Y., Kobayashi, S. e Tsuchida, K., “Block Diagram Generation and Parsing Base don Graph Grammar,” In *IEEE International Symposium on Circuits and Systems*, pages 1760-1763, Jun 1997.
- [8] Argewala, T., Choed-Amphai Y., “A Systhesis rule for concurrent systems,” In *Design Automation Conference (DAC’78)*, pages 305-311, June 1978.
- [9] Arlat, J., Kanoun, K., Laprie, J.C., “Dependability Evaluation of Software Fault-Tolerance”, In *Twenty-Fifth International Symposium on Fault-Tolerant Computing, Highlights from Twenty-Five Year*, pages 194-199, Jun 1995.
- [10] Arlat, J., Kanoun, K., Madeira, et al., “State of the Art,” DBench (Dependability Benchmarking) project, IST-2000-25425, LAAS – CNRS, France, August 2001.
- [11] Avizienis, A. and J. P. J. Kelly, “Fault Tolerance by Design Diversity: Concepts and Experiments,” *IEEE Computer*, pages 67-80, August, 1984.
- [12] Avizienis, A., “*The Methodology of N-Version Programming*,” In M.R. Lyu (ed.) Chapter 2 of *Software Fault Tolerance*, , Wiley, pages 23-46, 1995.
- [13] Avizienis, A., Laprie, J.C., Randell, B., “Fundamental Concepts of Dependability”, *Rapport LAAS N°01145*, 19p, April 2001.
- [14] Avizienis, A., Laprie, J.-C., Randell, B., "Dependability and its Threats: A TaxoNomy", In R. Jacquart (ed.) *Building the Information Society: Proc. IFIP 18th World Computer Congress, 22-27 August 2004, Toulouse, France*, pages 91-120, Kluwer Academic Publishers, 2004.
- [15] Balbo, G., “Introduction to stochastic Petri nets,” In J.-P. Katoen, H. Brinksma, and H. Hermanns (eds.), *Lectures on Formal Methods and Performance Analysis: First EEF/Euro Summer School on Trends in Computer Science* Berg en Dal, The Netherlands, July 3-7, 2000, Revised Lectures, Volume

- 2090 of Lecture Notes in Computer Science, pages 84--155. Springer-Verlag, 2001.
- [16] Béounes, C., et al., "SURF-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems," In *23rd Int. Symp. On Fault-Tolerance Computing*, pages 668-673, Toulouse (France), 1993.
 - [17] Bernardi, S., Bobbio, A., Donatelli, S., "Petri Nets and Dependability", In W. Reisig and G. Rozenberg (eds.), *Lectures on Concurrency and Petri Nets*, Springer Verlag - LNCS, Vol. 3098, pages 125-179, 2004.
 - [18] Bhat, U.N., "*Elements of Applied Stochastic Processes*," Wiley, New York, 1984.
 - [19] Blum, A.M., Goyal, A., Heidelberger, P., Lavenberg, S.S., Nakayama, M.K., Shahbuddin, P., "Modeling and analysis of system dependability using the System Availability Estimator," In *Twenty-Fourth International Symposium on Fault-Tolerant Computing FTCS-24*, Digest of Papers, pages 137-141, Jun 1994.
 - [20] Bobbio, A., Trivedi, K.S., "An Aggregation Technique for the Transient Analysis of Stiff Markov Chains," In *IEEE Trans. Computers*, Vol. 35, No. 9, pages 803-814, Sept. 1986.
 - [21] Bobbio, A., "System Modelling with Petri Nets", In A.G. Colombo and A.S. de Bustamante (eds.), *System Reliability Assessment*, Kluwer p.c., pages 102-143, 1990.
 - [22] Bolch, G., Greiner, S., de Meer, H., Trivedi, K., "*Queueing Networks and Markov Chains*", John Wiley & Sons, 1998.
 - [23] Bondavalli, A., Nelli, M., Simoncini, L., Mongardi, G., "Hierarchical modelling of complex control systems: dependability analysis of a railway interlocking", In *Computer Systems Science & Engineering*, Vol. 16, No. 4, pages 249-261, 2001.
 - [24] Brogan, W.L., "*Modern Control Theory*", Prentice-Hall, Englewood Cliffs, 1985.
 - [25] Bryant, R.E., "Graph Based Algorithms for Boolean Function Manipulation," In *IEEE Trans. Computers*, Vol. 35, No. 8, pp. 677-691, Aug. 1986.
 - [26] Buchacker, K., "Modeling with Extended Fault Trees," In *Proc. 5th Int. IEEE High-Assurance Systems Engineering Symposium*, Albuquerque, NM, pages 238-246, November 2000.
 - [27] Buchholz, P., "Exact and Ordinary Lumpability in Finite Markov Chains," In *J. Applied Probability*, Vol. 31, pages 59-74, 1994.
 - [28] Buchholz, P., Ciardo, G., Donatelli, S., Kemper, P., "Complexity of Memory-Efficient Kronecker Operations with Applications to the Solution of Markov Models," In *INFORMS J. Computing*, Vol. 12, No. 3, pages 203-222, 2000.
 - [29] Careem, S., Ravindran N., "A Survey of Specification Techniques for Modelling Fault Tolerant Degradable Computer Systems", In Final Report, 1993
 - [30] Carter, W.C. "Basic Concepts in Hardware Architecture for Reliable Computing", 2nd Advanced Course – Computing Systems Reliability, pages 10-21, September 1979.

- [31] Cassandras, C. G., Lafortune, S., “*Introduction to Discrete Event Systems*”, Kluwer Academic Publishers, Norwell, Massachusetts, 1999.
- [32] Chen, L., Avizienis, A., “N-version programming: A fault tolerant approach to reliability of software operation”, In *Proceedings of the International Symposium on Fault Tolerant Computing*, Toulouse, pages 3-9, June 1978.
- [33] Chiola, G., “GreatSPN 1.5 software architecture,” In *Proc. Fifth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 117-132, Torino, Italy, 1991.
- [34] Ciardo, G., Muppala, J.K., Trivedi, K.S., “SPNP: Stochastic Petri Net Package”, In *Proc. Int. Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, Los Alamitos, CA, pages 142-150, Dec. 1989.
- [35] Ciardo, G., Muppala, J.K., Trivedi, K.S., "Analyzing concurrent and fault-tolerant software using stochastic reward nets," In *Journal Parallel Distributed Computing*, Vol. 15, No. 3, pages 255-269, 1992.
- [36] Ciardo, G., Blakemore, A., Chimento, P.F., Muppala, J., Trivedi, K.S., “*Automated generation and analysis of Markov reward models using stochastic reward nets*,” In C. Meyer and R.J.Plemmons (eds), *Linear Algebra, Markov Chains, and Queuing Models*, IMA Volumes in Mathematics and its Applications, Vol. 48, pages 145-191, Springer-Verlag Heidelberg, Germany, 1993.
- [37] Ciardo, G., Lindemann, C., “Analysis of Deterministic and Stochastic Petri Nets”, In *Proceedings 5th International Workshop on Petri Nets and Performance Models*, pages 160-169, October 1993.
- [38] Ciardo, G., “Petri Nets with Marking-dependent Arc Cardinality: Properties and Analysis,” In *Proceedings of the 15th International Conference on Application and Theory of Petri Nets*, pages 179-198, *Lectures Notes in Computer Science #815* – Springer Verlag, 1994.
- [39] Ciardo, G., Miner, A.S., “A Data Structure for the Efficient Kronecker Solution of GSPNs,” In *Proc. 8th Int. Workshop on Petri Nets and Performance Models (PNPM’99)*, pages 22-29, Saragossa, Spain, 1999.
- [40] Courtois, P., “*Decomposability: Queueing and Computer System Applications*,” Academic Press, New York, 1977.
- [41] Courtois, P.J., Semal, P., “Computable Bounds for Conditional Steady-State Probabilities in Large Markov Chains and Queueing Models,” In *IEEE J. Selected Areas in Comm.*, Vol. 4, No. 6, pages 926-937, Sept. 1986.
- [42] Cox, D.R., “A use of complex probabilities in the theory of stochastic processes,” In *Proceedings of the Cambridge Philosophical Society*, Vol. 51, pages 313-319, 1955
- [43] Desrochers, A. A., Al-Jaar, R.Y., “*Applications of Petri Nets in Manufacturing Systems: Modeling Control, and Performance Analysis*,” IEEE Press, Piscataway, NJ, 1995.
- [44] Deavours, D.D., Clark, G., Courtney, T., Daly, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P.G., “The Mo’bius Framework and Its Implementation,” In *IEEE Trans. Software Eng.*, Vol. 28, No. 10, pages 956-969, Oct. 2002.

- [45] DiCesare, F., Jeng, M.D., “Synthesis for Manufacturing Systems Integration,” In *Practice of Petri Nets in Manufacturing*, Chapman & Hall, Ltd., London, UK, pages 103-146, 1993.
- [46] Dugan, J., Bavuso, S.J., Boyd, M.A., “Fault Trees and Sequence Dependencies,” In *Proc. Reliability and Maintainability Symp.*, pp. 286-293, 1990.
- [47] Dugan, J., Lyu, M., “Dependability Modeling for Fault-Tolerant Software and Systems”, *Software Fault Tolerant*, M.R.Lyu, ed., Chichester: John Wiley & Sons, pages 109-138, 1995.
- [48] Esary, J.D., Proschan, F., “ *The Reliability of Coherent Systems*,” In Wilcox and Mann (eds.) *Redundancy Techniques for Computing Systems*, Washington, DC:Spartan Books, pages 47-61, 1962.
- [49] Eyman, E., “*Modeling, Simulation and Control*”, West Publishing, st. Paul, 1988.
- [50] Fernandes, S.M.M., Dissertação de Mestrado: “Estudo e Implementação do mecanismo de Tolerância a Falhas em Software por meio de Blocos de Recuperação”, Dezembro de 1995.
- [51] Fernandes, S.M.M. e Maciel, P.R.M., “A Modeling Methodology for Reliability Evaluation of Hw/Sw Co-Design: an Approach Based on DSPN and Fault Tolerance,” In *Proceedings of the Design, Analysis, and Simulation of Distributed Systems (DASD2003) conference of the Advanced Simulation Technology Conference (ASTC-2003)*, Orlando, USA, April 2003.
- [52] Fernandes, S.M.M. e Maciel, P.R.M., “Reliability Evaluation for Dependable Embedded System Specifications,” In *Proceedings of the First ACM & IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE'2003)*, Mont Saint-Michel, France, June 2003.
- [53] Fernandes, S.M.M. e Maciel, P.R.M., “Parameterized GSPN Model and Extended Dependability Block Diagram for Reliability Evaluation of Embedded System,” In *IEEE International Conference on System, Man, and Cybernetics*, Taipei, Taiwan, October 2006.
- [54] Fernandes, S. M. M., Maciel, P. R. M., Rosa, Campos, M.A., Arteiro, R. D., N. S.; Projeto R-TOOL: Relatório Final (Metodologia); Projeto de P&D firmado entre CESAR e CHESF, Outubro, 2006.
- [55] Fernandes S.M.M, Maciel, P. R. M., M. A., Rosa, Campos, N. S., Arteiro, R. D., “Projeto R-TOOL: Relatório Técnico Final da Metodologia para Avaliação da Confiabilidade e Disponibilidade da Infra-estrutura de Comunicação da CHESF,” Projeto de P&D firmado entre CESAR e CHESF, Novembro, 2006.
- [56] Fota, N., Kaâniche, M., KaNoun, K., “Dependability Evaluation of an Air Traffic Control Computing System,” In *IEEE Internation Computer Performance and Dependability Symposium (IPDS'98)*, p. 206, 1998.
- [57] Gajski, D., Vahid, F., Narayan, S., Gong, J., “*Specification and Design of Embedded Systems*” Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [58] Geffroy, J.C, Motet, G. “*Design of Dependable Computing Systems*”, Kluwer Academic Publishers, Netherlands, 2002.

- [59] German, R., Kelling, Ch., Zimmermann, A., Hommel, G., "TimeNET - A Toolkit for Evaluating Stochastic Petri Nets with Non-Exponential Firing Times," In *Journal of Performance Evaluation*, Elsevier, Netherlands, 1995.
- [60] German, R., "Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets," John Wiley & Sons, 2000.
- [61] Girault, C., Valk, R., "Petri Nets for Systems Engineering: A guide to Modeling, Verification, and Applications", Springer-Verlag, Berlin, 2003.
- [62] Goyal, A., Carter, W. C., de Souza e Silva, E., Lavenberg, S. S., Trivedi, K. S., "The System Availability Estimator," In *Proceedings of the 16 Annual International Symposium on Fault-Tolerant Computing*, pages 84-89, IEEE Computer Society, 1986.
- [63] Götz, N., Hermanns, H., Herzog, U., Mertsiotakis, V., Rettelbach, M., "Stochastic Process Algebras: Constructive Specification Techniques Integrating Functional Performance and Dependability Aspects," In F. Bacelli, A. Jean-Marie and I. Mitran (eds.) *Quantitative Modeling in Parallel Systems*, Springer Verlag 1995
- [64] Haverkort, B.R., "Markovian Models for Performance and Dependability Evaluation," In J.-P. Katoen, H. Brinksma, and H. Hermanns (eds.), *Lectures on Formal Methods and Performance Analysis: First EEF/Euro Summer School on Trends in Computer Science Berg en Dal, The Netherlands, July 3-7, 2000, Revised Lectures*, Volume 2090 of Lecture Notes in Computer Science, pages 84--155. Springer-Verlag, 2001.
- [65] Heddaya, A., Helal, A., "Reliability, Availability, Dependability and Performability: A User-centered View," Technical Report No. 97-011, Boston University, USA, December 1996.
- [66] Hein, A., Cin, M.D., "Performance and dependability evaluation of scalable massively parallel computer systems with conjoint simulation," In *ACM Transactions on Modeling and Computer Simulation (TOMACS): special issue on Web-based modeling and simulation*, Vol. 8, Issue 4, pages 333-373, Oct. 1998.
- [67] Herzog, U., "Formal Methods for Performance Evaluation," In J.-P. Katoen, H. Brinksma, and H. Hermanns, editors, *Lectures on Formal Methods and Performance Analysis: First EEF/Euro Summer School on Trends in Computer Science Berg en Dal, The Netherlands, July 3-7, 2000, Revised Lectures*, Volume 2090 of Lecture Notes in Computer Science, pages 84--155. Springer-Verlag, 2001.
- [68] Hiller, M., "Software Fault-Tolerance Techniques from a Real-Time Systems Point of View", Technical Report No 98-16, Chalmers University of Technology, Sweden, November, 1998.
- [69] Hollanda, G.M., "Normas e Terminologia na Área da Confiabilidade". V *Simpósio de Computadores Tolerantes a Falhas*. São José dos Campos, SP. Brasil. Outubro de 1993
- [70] Johnson, B.A., "Design and Analysis of Fault-Tolerant Digital Systems", Addison-Wesley Publishing Company, Inc. 1989.

- [71] Jones, C., Randell, B., "Dependable Pervasive Systems", technical report University Newcastle upon Tyne, Cyber Trust & Crime Prevention Project, June 2004.
- [72] Kanoun, K., Ortalo-Borel, M., "Dependability of Fault-Tolerant Systems – Explicit Modeling of the Interactions Between Hardware and Software Components," In *2nd International Computer Performance and Dependability Symposium (IPDS '96)*, pages 252-261, 1996.
- [73] Kanoun, k., Borrel, M., Morteveille, T., Peytavin, A., "Availability of CAUTRA, a subset of the French Air Traffic Control System", In *IEEE Transaction on Computers*, Vol. 48, No. 5, pages 528-535, May 1999
- [74] Kanoun, K., Ortalo-Borel, M., "Fault-tolerant system dependability-explicit modeling of hardware and software component-interactions," In *IEEE Transactions on Reliability*, Vol. 49, issue 4, pages 363-376, Dec. 2000.
- [75] Knight, J.C., NakaNo, L.G., "Software Test Techniques For System Fault-Tree Analysis", In *16th International Conference on Computer Safety, Reliability, and Security, SAFECOMP'97*, 1997.
- [76] Kumar, S., Grassmann, W., Billington, R., "A Stable Algorithm to Calculate Steady-State Probability and Frequency of a Markov System," In *IEEE Transactions on Reliability*, Vol. 36, No. 1, April 1987.
- [77] Lala, J. H., Harper, R. E., "Architectural principles for safety-critical real-time applications," In *Proc. of the IEEE*, Vol. 82, No. 1, pages 25-40, January, 1994.
- [78] Lala, P., "*Self-Checking and Fault Tolerant Digital Design*", Academic Press, 2001.
- [79] Lanus, M., Yin, L., Trivedi, K.S., "Hierarchical Composition and Aggregation of State-Based Availability and Performability Models", In *IEEE Transactions on Reliability*, Vol. 52, No. 1, pages 44-52, March 2003.
- [80] Laprie, J.C., "Dependable computing and fault tolerance: concepts and termiNology", In *Digest of FTCS-15*, pages 2-11, June 1985.
- [81] Laprie, J.C., et al., "Hardware- and Software-Fault-Tolerance: Definition and Analysis of Architectural Solutions", In *Proceedings of the 17th International Symposium on Fault-Tolerant Computing*, pages 116-121, 1987.
- [82] Laprie, J.C., Arlat, J., Beounes, C., KaNoun, K., "Definition and Analysis of Hardware and Software Fault-Tolerant Architectures," In *IEEE Computer*, Vol. 23, pages 39-51, (Special Issue on Fault Tolerant Systems) 1990.
- [83] Laprie, J.C., "Dependability: Basic Concepts and Associated TermiNology," In *Dependable Computing and Fault-Tolerant Systems*, Vol. 5, Springer-Verlag, 1992
- [84] Latronico, B., Martin, C., Koopman, P., "Analysing Dependability of Embedded Systems from the User Perspective", In *Workshop on Reliability in Embedded Systems (in conjunction with SRDS)*, October 2001.
- [85] Leveson, N.G., "*Safeware: System Safety and Computers*," Addison-Wesley Publishing Co., 1995.
- [86] Lindemann, C., "*Performance Modelling with Deterministic and Stochastic Petri Nets*," John Wiley and Sons, Chichester, England, 1998.

- [87] Logothetis, D., Trivedi, K.S., Puliafito, A., "Markov Regenerative Models," In *IEEE International Computer Performance and Dependability Symposium (IPDS'95)*, p. 0134, 1995
- [88] Malhotra, M., Reibman, A., "Selecting and Implementing Phase Approximations for Semi--Markov Models", In *Communications in Statistics: Stochastic Models*, Vol. 9, No. 4, pages 473--506, 1993.
- [89] Malhotra, M., Trivedi, K., "Power-Hierarchy of Dependability Model Types", In *IEEE Transaction on Reliability*, Vol. 43, No. 2, pages 493-502, September 1994.
- [90] Malhotra, M., Trivedi, K., "Dependability Modeling Using Petri-Net Based Models", In *IEEE Transactions on Reliability*, Vol. 44, No. 3, pages 428-440, Sept., 1995.
- [91] Marsan, A., Balbo, G., Conte, G., "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", In *ACM Transactions on Computer Systems*, Vol. 2, No. 2, pages 93-122, May 1984.
- [92] Marsan, A., Chiola, G., "On Petri Nets with Deterministic and Exponentially Distributed Firing Times," In G. Rozenberg (ed.), *Advances in Petri Nets 1986, Lecture Notes in Computer Science*, Vol. 266, pages 132-145, Springer, 1987.
- [93] Marsan, A., "Stochastic Petri nets: an elementary introduction," In G. Rozenberg, (ed.), *Advances in Petri Nets 1989, Lecture Notes in Computer Science*, Springer Verlag, 1990.
- [94] Marsan, A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G., "Modelling with Generalized Stochastic Petri Nets," Wiley Series in Parallel Computing, 1995.
- [95] Marsan, A., Bobbio, M., Donatelli, S., "Petri Nets in Performance Analysis: An Introduction," In W. Reisig, G. Rozenberg, *Lecture Notes in Computer Science, Vol. 1491: Lectures on Petri Nets I: Basic Models*, pages 211-256, Springer-Verlag, 1998.
- [96] Mazigh, B., Gresser, J., Simon, F., "GSPN modelling methods for performance and dependability evaluation of a real-life flexible manufacturing system," In *Proceedings 5th International Workshop on Petri Nets and Performance Models*, pages 290-299, Oct. 1993.
- [97] Mitra, S., Saxena, N., McCluskey, E., "A design Diversity Metric and Reliability Analysis for Redundant Systems", In *Intl Test Conf.*, pages 662-671, 1999.
- [98] van Moorsel, A.P.A., "Action Models: A Reliability Modeling Formalism for Fault-Tolerant Distributed Computing Systems", In *IEEE International Computer Performance and Dependability Symposium (IPDS'98)*, Durham, North Carolina, September, 1998.
- [99] Mupala, J., Sathaye, A.S., Howe, R.C., Trivedi, K.S., "Dependability Modeling of a Heterogeneous VAXcluster System Using Stochastic Reward Nets," In *Hardware and Software Fault Tolerance in Parallel Computing Systems*, Ellis Horwood Ltd., pages 33-59, 1992.

- [100] Muppala, J., Giardo, G., Trivedi, K.S., “Stochastic Rewards Nets for Reliability Prediction”, In *Communications in Reliability, Maintainability and Serviceability*, Vol. 1, No. 2, pages 9-20, July 1994.
- [101] Muppala, J., Fricks, R., Trivedi, K.S., “Techniques for System Dependability Evaluation” In W.Grassman, editor, *Computational Probability*, pages 445-480, Kluwer Academic, The Netherlands, 2000.
- [102] Mura, I., Bondavalli, A., Zang, X., Trivedi, K.S., “Dependability Modeling and Evaluation of Phased Mission Systems: a DSPN Approach,” In *7th IFIP Int. Conference on Dependable Computing for Critical Applications*, IEEE Society Press, pages 299-318, San Jose, CA, USA, 1999.
- [103] Mura, I., Bondavalli, A., “Herarchical Modeling & Evaluation of Phased-Mission Systems,” In *IEEE Transactions on Reliability*, Vol. 48, No. 4, December 1999.
- [104] Mura, I., Bondavalli, A., “Markov Regenerative Stochastic Petri Nets to Model and Evaluate Phased Mission Systems Dependability,” In *IEEE Trans. Computers*, Vol. 50, No. 12, pages 1337-1351, 2001.
- [105] Murata, T., “Petri nets: properties, analysis, and applications,” In *Proceedings of the IEEE*, Vol. 77, No. 4, pages 541-580, April 1989.
- [106] Nicol, D., Sanders, W., Trivedi, K.S., “Model-Based Evaluation: From Dependability to Security”, In *IEEE Trans. Dependable and Secure Computing*, Vol. 1, No. 1, Jan-March 2004.
- [107] Nicola, V., Nakajama, M., Heidelberg, P., Goyal, A., “Fast Simulation of Dependability Models with General Failure, Repair and Maintenance Processes,” In *Proc. 20th Annual Int. Symp. On Fault-Tolerant Computing Systems*, pages 491-498, Newcastle upon Tyne, UK, Los Alamitos, CA, IEEE Computer Society Press, June 1990.
- [108] Petri, C.A., “Kommunikation mit Automaten”, Dissertation, University of Bonn, Bonn, Germany, 1962.
- [109] Poledna, S., “*Fault Tolerant Real Time Systems: The Problem of Replica Determinism*”, Kluwer Academic Publishers, 1996.
- [110] Popstjanova, K.G., Grnarov, A., “Performability and Reliability Modeling of N Version Fault Tolerant Software in Real Time Systems,” In *23rd EUROMICRO Conference '97 New Frontiers of Information Technology*, 1997.
- [111] Porcarelli, S., Giandomenico, F.D., Lollini, P., Bondavalli, A., “A Modular Approach for Model-based Dependability Evaluation of a Class of Systems,” In *International Service Availability Symposium (ISAS)*, Munich, Germany, May 2004.
- [112] Powell, D., Cukier, M., Arlat, J., Crouzet, Y., “Estimation of time-dependent coverage” Rapport LAAS No. 96466, Contrat ESPRIT DeVa Project N°20072, pages 541-560, Décembre 1997.
- [113] Pradhan, D. K., “*Fault Tolerant Computer System Design*”, Prentice Hall Advanced Computing And Telecommunications Series, Upper Saddle River, NJ, 1996.
- [114] Prodan, L., Udrescu, M., Vladutiu, M., “A Dependability Perspective on Emerging Technologies,” In *Conf. Computing Frontiers*, pages 187-198, Ischia, Italy, May 2006.

- [115] Pullum, L.L., “*Software Fault Tolerance Techniques and Implementation*,” Artech House, 2001.
- [116] Rabah, M., Kanoun, K., “Performability Evaluation of Multipurpose Multiprocessor Systems: The “Separatio of Concerns” Approach,” In *IEEE Transactions on Computers*, Vol. 52, No. 2, February 2003.
- [117] Rai, S., Veeraraghavan, M., Trivedi, K.S., “A Survey on Efficient Computation of Reliability Using Disjoint Products Approach,” In *Networks*, Vol. 25, No. 3, pages 147-163, 1995.
- [118] Randell, B., “System structure for software fault tolerance”, In *IEEE Trans. On Software Engineering*, SE, Vol. 1, No. 2, pages 220-232, June 1975.
- [119] Reibman, A., Veeraraghavan, M., “Reliability modeling: An overview for system designers”, In *IEEE Computer*, pages 49-57, April 1991.
- [120] Ripley, B. D., “Stochastic Simulation,” Wiley & Sons, 237pages, 1987
- [121] Sahner, R., Trivedi, K.S., “ A Hierarchical, Combinatorial-Markov Method of Solving Complex Reliability Models,” In *Fall Joint Computer Conference, AFIPS*, pages 817-825, New York, 1986.
- [122] Sahner, R.A., Trivedi, K.S., Puliafito, A., “*Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*”, Kluwer Academic Publishers, 1996
- [123] Sanders, W. H., Obal II, W. D., Qureshi, M. A., Widjanarko, F. K., “The UltraSAN modeling environment. Performance Evaluation”, Vol. 21, 1995.
- [124] Sanders, W., Meyer, J., “Stochastic Activity Networks: Formal Definitions and Concepts”, In *Lecture Notes in Computer Science*, 2001
- [125] Schneeweiss, W.G., “*The Fault Tree Method*,” LiLoLe Verlag, 1999.
- [126] Siewiorek, D., Swarz, R., “*Reliable Computer Systems – Design and Evaluation*”, A. K. Peters, third Edition, 1998.
- [127] Siewiorek, D.P., Chillarege, R., Kalbarczyk, Z.T., “Reflections on Industry Trends and Experimental Research in Dependability,” In *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 2, April-June 2004.
- [128] Silva, M., “*Introducing Petri nets*,” In *Practice of Petri Nets in Manufacturing*, Chapman & Hall, Ltd., London, UK, pages 103-146, 1993.
- [129] Simoncini, L., Di Giandomenico, F., Bondavalli, A., Chiaradonna, S., “Architectural challenges for a dependable information society”, In *Fault Tolerance for Trustworthy and Dependable Information Infrastructures, Topical Days Track, WCC 18th IFIP World Computer Congress*, Toulouse, France, August, pages 22-27, 2004.
- [130] Singh, C., Billinton, R., Lee, S., “The method of stages for non-Markovian models,” *IEEE Transactions on Reliability*, R-26(1):135-137, June 1977.
- [131] Storey, N., (1999) “Design for Safety,” In *Proc. 7th Safety-Critical Systems Symposium*, pages 1-25, Huntingon, UK. Feb. 1999,.
- [132] SURF-2 User guide. LAAS-CNRS, 1994.
- [133] Takahashi, Y., “A Lumping Method for Numerical Calculations of Stationary Distributions of Markov Chains,” Research report B 18, Tokyo Institute of Technology, Department of Information Sciences, Tokyo, June 1975.

- [134] TimeNet3.0 Users Manual, “A Software Tool for the Performability evaluation with Stochastic Petri Nets”, TU Berlin, June 2001.
- [135] Tomek, L., Mainkar, V., Geist, R., Trivedi, K., “Reliability Modeling of Life-Critical, Real-Time Systems”, In *Proceedings of the IEEE*, Vol. 82, No. 1, pages 108-121, January 1994.
- [136] Trivedi. K., “*Probability and Statistics with Reliability, Queueing, and Computer Science Applications*”, 1st edition, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [137] Trivedi, K.S., Malhotra, M., “Reliability and Performability Techniques and Tools: A Survey,” In *MMB*, pages 27-48, 1993.
- [138] Trivedi, K. S., Ciardo, G., Mahorta, M., Sahner, R. A.,”Dependability and performability analysis.” In *Performance evaluation of computer and communication systems: Joint tutorial papers of Performance '93 and Sigmetrics '93*, L. Donatelli and R. Nelson (eds.), Springer, LNCS 729. pages 587—612, 1993
- [139] Trivedi, K., Hunter, S., Garg, S., Fricks, R.,“Reliability Analysis Techniques Explored Through a Communication Network Example”, In *International Workshop on Computer-Aided Design, Test, and Evaluation for Dependability*, Beijing, China, July 2-3, 1996.
- [140] Trivedi. K., “*Probability and Statistics with Reliability, Queueing, and Computer Science Applications*”, 2nd edition, John Wiley & Sons, Inc., New York, 2002.
- [141] U. S. Department of Defense, *Electronic Reliability Design Handbook*, U.S. Military Handbook MIL-HDBK-338B, 1998.
- [142] Vallete, R. “Analysis of Petri nets by stepwise refinement”, *Journal of Computer Systems Science*, Vol. 18, pages 35-46, 1979.
- [143] Von Neumann, J., “Probabilistic logics and synthesis of reliable organisms from unreliable components”, *Automata Studies*, In *Annals of Mathematical Studies* No. 34, eds. C.E.ShanNon and J. McCarthy, 43-98, Princeton University Press, 1956.
- [144] Wang, W., Loman, J.M., ArNo, R.G., Vassiliou, P., Furlong, E.R. e Ogden, D., “Reliability Block Diagram Simulation Techniques Applied to the IEEE Std. 493 Standard Network,” In *IEEE Transactions on Industry Applications*, Vol. 40, No. 3, pages 887-895, May/June 2004.
- [145] Zimmermann, A., German, R., Freiheit, J., Hommel, G., “TimeNET 3.0 Tool Description,” In *International Conference on Petri Nets and Performance Models*, Zaragoza, Spain, 1999.

ANEXO-A

Processos de Nascimento e Morte

Por meio dos processos de nascimento e morte são definidas expressões de disponibilidade para sistemas paralelos redundantes, em função do número de reparadores. Por meio da notação de Kendall, os processos de nascimento e morte são descritos de um modo conciso. A notação de Kendall é caracterizada através de 4 parâmetros básicos:

$A / B / m$ – disciplina da fila, conforme a Tabela A.1, onde m é o número de reparadores.

Tabela A.1 Notação de Kendall

Tipos de Distribuição dos parâmetros A / B	Disciplina da Fila
C_k – Cox c/k fases	FCFS (Primeiro a chegar Primeiro a ser reparado).
D – Determinística	Prioridades Estáticas
E_k – Erlang c/k fases	Prioridades Dinâmicas
G – Geral	Interrupção ou Preempção
GI – Geral c/tempos de chegada independentes	RR (Round Robin)
H_k – Hyperexponencial c/k fases	PS (Processor Sharing)

Existem outras disciplinas de fila que não estão aqui especificadas. Relatamos na Tabela A.1 apenas aquelas que poderão ser objetos de verificação para a compreensão dos modelos de dependabilidade. Os processos de nascimento e morte [140] a seguir podem ser utilizados para descrição das expressões de disponibilidade em função do número de reparadores, além do estabelecimento das estratégias de reparo dos blocos que compõem um diagrama de blocos de sistema. Considerando-se um sistema composto por uma série de blocos em uma configuração modular, tem-se:

A.1 Processo Especial de Nascimento e Morte: Rede de Fila M/M/m

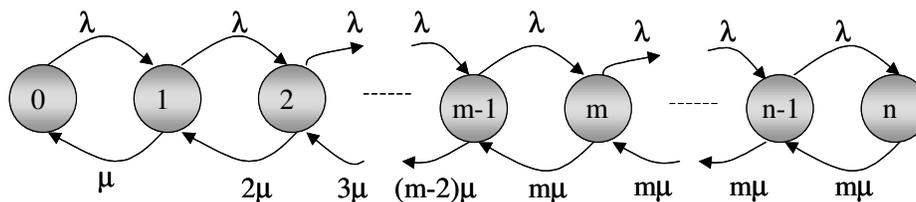


Figura A.1 Diagrama de estados da rede de fila M/M/m

Neste processo, os blocos de um diagrama de blocos falham seqüencialmente, isto é, um por vez, enquanto o reparo é realizado por até m reparadores simultaneamente (em paralelo). Este processo é mostrado na Figura A.1.

Características:

- λ : taxa de falha de cada bloco, considerando que todos tenham distribuições semelhantes;
- μ : taxa de reparo de cada bloco, considerando que todos os reparadores tenham a mesma taxa de reparo;
- n : número de componentes em falha no sistema em determinado instante de tempo;
- Política de reparo: de acordo com a ordem de chegada à manutenção, ou seja, FCFS;
- N° de reparadores: m .

Taxas:

- $\lambda_k = \lambda$, $k=0, 1, 2, \dots$
- $\mu_k = k\mu$, para $0 \leq k < m$,
- e
- $\mu_k = m\mu$, para $m \leq k$.

Probabilidades de estado permanente:

- $$\pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu} \right)^k \frac{1}{k!}, \quad \text{para } k < m,$$

$$\pi_k = \pi_0 \prod_{i=0}^{m-1} \frac{\lambda}{(i+1)\mu} \prod_{j=m}^{k-1} \frac{\lambda}{m\mu} = \pi_0 \left(\frac{\lambda}{\mu} \right)^k \frac{1}{m! m^{k-m}}, \quad \text{para } m \leq k \leq n,$$

Definindo-se a intensidade de tráfego por $\rho = \lambda/(m\mu)$, e $\sum_{k=0}^n \pi_k = 1$, tem-se a seguinte expressão:

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}$$

As expressões obtidas através da fila M/M/m são utilizadas nos modelos em redes de Petri parametrizadas para avaliação de disponibilidade dos blocos utilizando-se técnicas de redundância do tipo *coldstandby*, com estratégia de reparo múltipla e política de reparo FCFS.

A.2 Processo Especial de Nascimento e Morte: Reparador Único

Neste processo vários blocos de um diagrama de sistema estão ativos simultaneamente e concorrentemente, podendo falhar na forma $(M-j)\lambda$, onde $j = 0, 1, 2, \dots, M$, conforme a Figura A.2.

Características:

- λ : taxa de falha de cada bloco considerando que todos tenham a mesma distribuição;
- μ : taxa de reparo de cada bloco;
- M : número de componentes ativos no sistema em determinado instante de tempo;
- j : número de componentes em falha em um determinado instante de tempo;
- Política de reparo: de acordo com a ordem de chegada à manutenção, ou seja, FCFS;
- N° de reparadores: $m=1$.

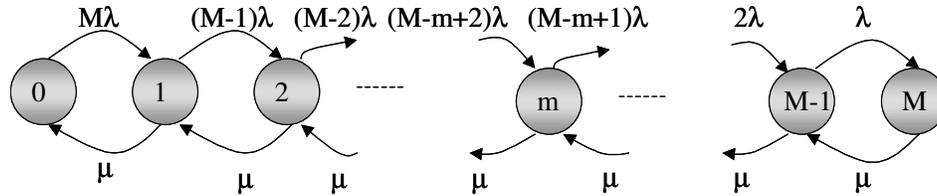


Figura A.2 Diagrama de estados do modelo *machine repairman*

Probabilidades de estado permanente:

$\pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda(M-i)}{\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{M!}{(M-k)!} = \pi_0 \rho^k \frac{M!}{(M-k)!}$, onde $\rho = \lambda/\mu$ é a intensidade de tráfego neste tipo de processo.

$$\pi_0 = \frac{1}{\sum_{k=0}^M \rho^k \frac{M!}{(M-k)!}}$$

Neste tipo de processo os blocos do sistema estão em execução simultânea (paralela), porém com estratégia de reparo única e política de reparo FCFS.

A.3 Processo Especial de Nascimento e Morte: Independência estocástica

Neste tipo de processo todos os blocos componentes estão em atividade simultaneamente, contudo em caso de falha são prontamente atendidos por reparadores individuais na mesma quantidade, conforme pode ser observado na Figura A.3.

Características:

- λ : taxa de falha de cada bloco (mesma distribuição);
- μ : taxa de reparo de cada bloco (individual para cada bloco);
- M : número de componentes ativos no sistema em determinado instante de tempo;
- j : número de componentes em falha em um determinado instante de tempo;
- Política de reparo: de acordo com a ordem de chegada à manutenção, ou seja, FCFS;

- N° de reparadores: M.

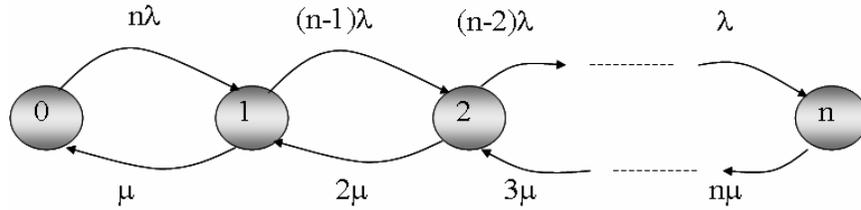


Figura A.3 Diagrama do modelo com independência estocástica

Taxas:

- $\lambda_k = k\lambda$, $k=0, 1, 2, \dots, M$
- $\mu_k = k\mu$, $k=0, 1, 2, \dots, M$

Probabilidades de estado permanente:

Considerando-se as equações de balanceamento e simplificações, tem-se:

$$\pi_M = \frac{(M)\lambda(M-1)\lambda(M-2)\lambda\dots(3)\lambda(2)\lambda\lambda}{\mu(2)\mu(3)\mu\dots(M-2)\mu(M-1)\mu(M)\mu} \therefore \pi_M = \left(\frac{\lambda}{\mu}\right)^M \pi_0.$$

A expressão obtida, conjuntamente com $\sum_{k=1}^M \pi_k = 1$ definem as probabilidades de estado permanente.

$$\pi_M = \left(\frac{1}{1 + \frac{\mu}{\lambda}} \right)^M$$

A expressão algébrica acima define a indisponibilidade do sistema composto por M blocos estocasticamente independentes. A disponibilidade do mesmo sistema, considerando-se a intensidade de tráfego dada por $\rho = \lambda/\mu$, é dada pela expressão:

$$A = 1 - \pi_M = 1 - \left(\frac{1}{1 + \frac{1}{\rho}} \right)^M = 1 - \left(\frac{\rho}{\rho + 1} \right)^M$$

As expressões obtidas através do modelo de independência estocástica são utilizadas nos modelos em redes de Petri parametrizadas para avaliação de disponibilidade dos blocos utilizando-se técnicas de redundância do tipo *hotstandby*, com estratégia de reparo múltipla e política de reparo FCFS.

ANEXO-B

Sintaxe da Ferramenta de Modelagem

A sintaxe das expressões dependentes das marcações utilizadas por arcos e retardos das transições temporizadas, na descrição dos modelos GSPN e MDP, através da ferramenta de modelagem TimeNET 3.0 é definida a seguir:

B.1 Símbolos Utilizados

<i>“symbol”</i>	símbolo terminal
<symbol>	símbolo não terminal
<i>expr</i>	expressão
<i>expr1 expr2</i>	expressão 1 ou expressão 2
{ <i>expression</i> }	qualquer número de ocorrências da expressão (incluindo nenhum)
<md_exp_delay>	retardo dependente da marcação
<md_weight>	peso de disparo de uma transição imediata dependente da marcação
<md_enable>	função de guarda de uma transição imediata dependente da marcação
<md_arc_mult>	multiplicidade de arco de entrada, de saída ou inibidor dependente da marcação
<reward_def>	definição de métrica
<param_def>	definição de parâmetro de retardo dependente de outros parâmetros de retardo

B.2 Definição da Sintaxe

<md_exp_delay>	:<if_expr>
<md_weight>	:<if_expr>
<md_enable>	:<logic_condition>”;
<md_arc_mult>	: <if_expr>
<reward_def>	:<expression>”;
<param_def>	: <expression>”;
<if_expr>	: { “IF” <logic_condition> ”:” <expression> } “ELSE” <expression> “;” <expression> “;”
<expression>	: <real_value> “-“<expression> “(“ <expression> “)” <expression> <num_op> <expression>
<real_value>	: <real_parameter> <real_constant>

	<real_item> (<reward_def> only)
	<integer_value>
<real_parameter>	: <identifier>
<real_constant>	: <digit> { <digit> } "." { <digit> } [<exponent>] { <digit> } "." <digit> { <digit> } [<exponent>] <digit> { <digit> } <exponent>
<exponent>	: ("E" "e") ("+" "-" "") <digit> { <digit> }
<rew_item>	: "P{ " <logic_condition> " }" "P{ " <logic_condition> "IF" <logic_condition> " }" "E{ " <marc_func> " }" "E{ " <marc_func> "IF" <logic_condition> " }"
<logic_condition>	: <comparison> "NOT" <logic_condition> "(" <logic_condition> ")" <logic_condition> "OR" <logic_condition> <logic_condition> "AND" <logic_condition>
<comparison>	: <mark_func> <comp_oper> <mark_func>
<comp_oper>	: "=" "/=" ">" "<" ">=" "<="
<mark_func>	: <mark_func> <num_op> <mark_func> "(" <mark_func> ")" <integer_value>
<num_op>	: "+" "-" "*" "/" "^"
<integer_value>	: <integer_constant> <integer_parameter> <marking> (não dentro da definição de parâmetro)
<integer_constant>	: <digit> { <digit> }
<integer_parameter>	: <identifier>
<marking>	: # <place_name>
<place_name>	: <identifier>
<identifier>	: <letter> { <letter> <digit> }
<letter>	: "a" .. "z" "A" .. "Z"
<digit>	: "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"

Os elementos especiais para definição de métricas são explicados a seguir. Todos eles mostram o resultado da execução da análise ou simulação, em estado permanente ou em um instante de tempo transiente.

"P{ " <logic_condition> " }"
= Probabilidade da <logic_condition>

"P{ " <logic_condition_1> "IF" <logic_condition_2> " }"
= Probabilidade da <logic_condition_1> sob a
precondição da <logic_condition_2> (probabilidade condicional)

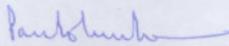
"E{ " <marc_func> " }"
= valor esperado da expressão dependente da marcação <marc_func>

“E{“ <marc_func> ”IF” <logic_condition> ”}”

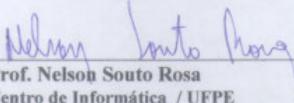
= valor esperado da expressão dependente da marcação

<marc_func>; somente na marcações onde <logic_condition>
avaliadas como verdadeiras são levadas em consideração

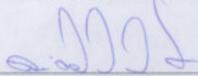
Tese de Doutorado apresentada por **Sérgio Murilo Maciel Fernandes** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "Avaliação de Dependabilidade de Sistemas com Mecanismos Tolerantes a Falha: Desenvolvimento de um Método Híbrido Baseado em EDSPN e Diagrama de Blocos", orientada pelo Prof. Paulo Romero Martins Maciel e aprovada pela Banca Examinadora formada pelos professores:



Prof. Paulo Roberto Freire Cunha
Centro de Informática / UFPE



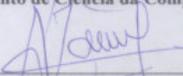
Prof. Nelson Souto Rosa
Centro de Informática / UFPE



Prof. Ricardo Massa Ferreira Lima
Escola Politécnica de Pernambuco / UPE

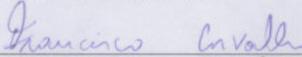


Prof. Antonio Otavio Fernandes
Departamento de Ciência da Computação / UFMG



Prof. Zair Abdelouahab
Departamento de Engenharia Elétrica / UFMA

Visto e permitida a impressão.
Recife, 27 de fevereiro de 2007.



Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.