VOLNYS BORGES BERNAL

MODELAMENTO DA IMPRECISÃO TEMPORAL DA OBSERVAÇÃO EM SISTEMAS DE DIAGNÓSTICO DE AMBIENTES DISTRIBUÍDOS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia

São Paulo

2003

VOLNYS BORGES BERNAL

MODELAMENTO DA IMPRECISÃO TEMPORAL DA OBSERVAÇÃO EM SISTEMAS DE DIAGNÓSTICO DE AMBIENTES DISTRIBUÍDOS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia

Área de concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Sergio Takeo Kofuji

São Paulo

2003

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 28 de abril de 2003.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Bernal, Volnys Borges
Modelamento da imprecisão temporal da observação em sistemas de diagnóstico de ambientes distribuídos / Volnys Borges Bernal. -- ed.rev. -- São Paulo, 2003.
207p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Diagnóstico baseado em modelo 2.Redes de computadores (Gerenciamento) 3.Aquisição de conhecimento. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

RESUMO

Esta tese trata a respeito de diagnóstico temporal. O diagnóstico temporal permite a utilização de informações a respeito da localização temporal (instantes de início, término e duração) de uma ocorrência. Não somente os valores associados aos objetos gerenciados, mas também a localização temporal destas ocorrências podem ser analisados, fornecendo informações adicionais à tarefa de diagnóstico. Porém, de nada vale a observação se a informação temporal nela contida for imprecisa. E, principalmente em redes de dados, esta imprecisão é intrínseca à dinâmica da observação, não sendo possível eliminá-la em diversas situações. É apresentado um método de modelamento da observação que permite deixar explícito tais imprecisões. Além do método, todo o processo de observação foi modelado, possibilitando também o entendimento de outros problemas associados às observações. Para mostrar a valia do modelamento da observação foi proposto um método de diagnóstico temporal que utiliza tais informações.

ABSTRACT

This thesis is about temporal diagnosis. The temporal diagnosis allows to handle information about temporal location (beginning, ending and duration) of an occurrence. It is possible to analyze not only values associated to the managed object, but also the temporal location of these occurrences, providing additional information to the diagnosis task. However, the observation is useless if the temporal information inserted to it is inaccurate. And, mainly in data networks, this imprecision is intrinsic to the observation dynamics, and it is not possible to eliminate it in several situations. A observation modeling method is presented that allows to explicit such imprecision. Besides of this method, all the observation process was shaped, also making possible the understanding of other problems associated to the observations. To show the validation of the observation modeling a temporal diagnosis method that uses such information was proposed.

Sumário

Sumário iii

Lista de figuras xii

Lista de tabelas xix

Lista de Definições xx

Lista de Abreviaturas xxii

1. Introdução 1

1.1 Detecção de problemas 2

1.1.1 Sistemas de apoio 4

1.1.2 Gerenciamento 4

1.1.3 Console de alarmes das plataformas de gerenciamento 6

1.1.4 Sistemas de correlação de alarmes 6

1.1.5 Sistemas de diagnóstico 7

1.2 Motivação 8

1.3 Objetivo 11

1.4 Justificativa 11

1.5 Trabalhos Relacionados 12

1.6 Estrutura da tese 14

2. Sistemas de Correlação e Diagnóstico 16

2.1 Sistemas de correlação de eventos 16

2.2 Sistemas de diagnóstico 18

2.3 Sistemas de diagnóstico 19

2.4 Classes de sistemas de diagnóstico 20

2.4.1 Sistemas de diagnóstico baseados em heurística 20

2.4.2 Sistemas de diagnóstico baseados em modelo 21
2.5 Sistemas de Diagnóstico Baseados em Modelo 23
2.5.1 Classificação de sistemas de diagnóstico 23
2.5.2 Classificação quanto aos modelos utilizados 24
2.5.2.1 A utilização de múltiplos modelos 31
2.5.3 Classificação quanto ao tipo de modelo de funcionamento 31
2.5.4 Classificação quando a ser quantitativo ou qualitativo 32
2.5.5 Classificação quanto ao tipo de inferência 33
2.5.6 Classificação quanto a forma de modelamento temporal 34
2.6 Estudo de caso: O sistema SMARTS 34
2.6.1 Modelo 37
2.6.2 Método de correlação por livro-código (*codebook*) 39
2.6.3 Principais vantagens 43
2.6.4 Principais desvantagens 44
2.7 Conclusão 44
3. Sistemas de diagnóstico baseado em modelo com dimensão temporal 45
3.1 Modelagem temporal em sistemas baseados em modelo 45
3.1.1 Diagnóstico atemporal sobre um único instante 46
3.1.2 Diagnóstico atemporal sobre coleção de sintomas 46
3.1.3 Diagnóstico atemporal sobre múltiplos instantes 47
3.1.4 Diagnóstico temporal 48
3.1.5 Diagnóstico temporal variante no tempo 49
3.2 Ontologia do tempo 50
3.2.1 Tempo quantitativo 51
3.2.2 Tempo qualitativo 51

3.2.3 Tempo como uma seqüência de estados....51

3.2.4 Abstrações *Ad hoc*....51

4. Interação do sistema de diagnóstico com o ambiente....52

4.1 Classificação quanto ao tipo da observação recebida....52

4.1.1 Observação tipo valor bruto (*raw value*)....54

4.1.2 Observação tipo estado....55

4.1.3 Observação tipo transição de estado....55

4.1.4 Observação tipo intervalo de tempo de ocorrência de estado....56

4.2 Classificação quanto ao controle de processo de observações....57

4.2.1 Passivo puro....57

4.2.2 Ativo puro....57

4.2.3 Semi-ativo....58

4.3 Classificação quanto ao momento do diagnóstico....58

4.3.1 Diagnóstico de momento corrente (DMC)....59

4.3.2 Diagnóstico de momento passado (DMP)....59

4.3.3 Diagnóstico contínuo (DC)....60

5. Capítulo 5 - Anomalias, sintomas e suas relações....61

5.1 Do objeto gerenciado ao sintoma....61

5.1.1 Componente....61

5.1.2 Anomalia....62

5.1.3 Objeto gerenciado....62

5.1.4 Objeto intermediário....67

5.1.5 Observação....70

5.1.6 Sintoma....71

5.1.7 Exemplo....72

5.2 Caracterização de uma relação causal 73
5.2.1 Classificação quanto à possibilidade de causar um efeito 73
5.2.2 Classificação quanto ao retardo do efeito 74
5.2.3 Relação causal entre anomalias e entre anomalia e sintoma 76
5.3 Relacionamento causal direto entre anomalias 76
5.4 Relacionamento causal direto entre anomalia e sintoma 78
5.4.1 Relacionamento causal indireto entre sintomas 80
5.4.2 Relacionamento causal direto entre sintomas 81
5.5 Conclusão 82
6. Proposta de modelagem da imprecisão temporal das observações 83
6.1 Caracterização temporal da observação quanto à sua defasagem 84
6.1.1 Observação não defasada 85
6.1.2 Observação defasada em até 1 ciclo 86
6.1.3 Observação defasada em até 2 ciclos 87
6.1.3.1 Situação A 88
6.1.3.2 Situação B 88
6.2 Problemas temporais de uma observação 89
6.2.1 Defasagem de tempo na observação do estado de um objeto gerenciado 89
6.2.2 Estado atual do objeto gerenciável disponível somente na próxima observação 90
6.2.3 Relacionamento de observações defasadas no tempo 90
6.2.3.1 Observações defasadas devido à utilização de períodos de amostragem diferentes 91
6.2.3.2 Observações defasadas devido à utilização de fases de amostragem diferentes 93
6.3 Modelamento das incertezas temporais associadas às observações 94

6.3.1 A nova representação da observação....94
6.3.2 O modelo tradicional de geração de observações....96
6.3.3 O novo modelo para geração de observações....97
6.4 Modelamento da observação: Intervalos de possibilidade e certeza....98
6.4.1 Modelamento de observações não defasadas....99
6.4.2 Modelamento de observações defasadas em até 1 ciclo....99
6.4.3 Modelamento de observações defasadas em até 2 ciclos....100
6.4.4 Trabalhando com a completude do conjunto de estados....101
6.5 Modelamento da observação: intervalos de incerteza....102
6.5.1 Intervalo de incerteza gerado pela perda de observação....103
6.5.1.1 Perda de eventos assíncronos....103
6.5.1.2 Perda de amostragens....103
6.5.2 Intervalo de incerteza gerado pela proximidade ao instante corrente....105
6.5.2.1 Modelamento para observação não defasada....105
6.5.2.2 Modelamento para observação defasada em até 1 ciclo....105
6.5.2.3 Modelamento para observação defasada em até 2 ciclos....106
6.5.3 Intervalo de incerteza existente no início do processo de monitoração....108
6.5.3.1 Nas observações defasadas em até 1 ciclo....108
6.5.3.2 Nas observações defasadas em até 2 ciclos.....108
6.6 Modelamento da observação: atrasos de comunicação....109
6.7 Algoritmo para modelamento da observação....110
6.7.1 Modelamento de observação não defasada....112
6.7.2 Modelamento para observações defasadas em até 1 ciclo....112
6.7.3 Modelamento para observações defasadas em até 2 ciclos....115
6.8 Aglomerado (*cluster*) de intervalos....119

6.9 Forma normal para *cluster* 121

6.9.1 O processo de normalização 121

6.10 Intersecção temporal entre *clusters* na forma normal 122

6.11 Intersecção temporal em relações causais entre anomalia e sintoma 123

6.12 Intersecção temporal em relações causais entre anomalias 124

6.13 Conclusão 125

7. Modelos e Métodos para diagnóstico temporal 126

7.1 SiDiR-t 126

7.2 Diagnóstico temporal utilizado no SiDiR-t 127

7.3 Modelos utilizados 127

7.4 Método de diagnóstico 131

7.5 Descrição do método de diagnóstico 132

7.5.1 Detecção de sintomas 133

7.5.2 Geração de hipóteses 135

7.5.2.1 Achar contribuintes 137

7.5.2.2 Transformar em conjunto de hipóteses 137

7.5.2.3 Filtragem baseada em predição 138

7.5.3 Discriminação de hipóteses 138

8. Conclusão 140

8.1 Conclusões 141

8.2 Contribuições 141

8.3 Limitações 143

8.4 Trabalhos futuros 143

Anexo 1. Gerenciamento de Redes 145

1 Áreas funcionais do gerenciamento 146

2 Modelo de gerenciamento 147
3 Protocolos de gerenciamento 148
4 SNMP 149
4.1 SNMP v1 149
4.2 SNMP v2 150
4.3 SNMP v3 151
4.4 Gerenciamento distribuído 151
4.4.1 MIBS RMONx 151
4.4.2 “Expression MIB” e “Event MIB” 151
4.4.3 “Script MIB” 151
4.4.4 “Ping”, “traceroute” e “nslookup” remoto 152
5 DMI 152
6 Gerenciamento OSI 153
6.1 Comunicação de gerenciamento entre sistemas 154
6.2 Elementos de serviço do protocolo CMIP 155
6.3 Estrutura da informação de gerenciamento 156
6.3.5 Árvore de herança 156
6.3.6 Árvore de nomeação 157
6.3.7 Árvore de registo 158
6.3.8 Escopo 158
6.3.9 Filtro 159
7 TMN (Padrão OSI para telecomunicação) 159
7.1 Gerenciamento de redes de telecomunicações 161
7.2 Padrão TMN 162
7.3 Modelo de camadas de gerenciamento 163

7.3.10 Camada de elemento de rede 164
7.3.11 Camada de gerenciamento de elemento de rede 165
7.3.12 Camada de gerenciamento de rede 165
7.3.13 Camada de gerenciamento de serviços 165
7.3.14 Camada de gerenciamento de negócios 165
Anexo 2. Relações causais 167
1 Relação causal 167
2 Formas de representação da relação causal 169
3 Grafo de correlação 170
4 Modelos de causalidade 171
Anexo 3. Código de Hamming 173
1 Códigos de repetição 173
2 Código de bloco binário 173
3 Redundância 174
4 Syndrome 174
5 Código de Hamming 175
6 Distância do código de Hamming 175
Anexo 4. Representação de tempo 176
1 Ponto de Tempo 177
1.1 Relações 178
2 Intervalo de Tempo Convexo 178
2.1 Relações primitivas entre intervalos de tempo convexo 180
2.2 Relações adicionais entre intervalos de tempo convexo 181
2.3 Operadores sobre intervalos de tempo convexo 182
2.4 Transitividade 183

3 Relações entre intervalos e pontos de tempo.....186

4 Conclusão 186

Anexo 5. Exemplos de modelamento de observação 187

1 Exemplo #1 – Defasada em até 1 ciclo 187

2 Exemplo #2 – Defasada em até 2 ciclos.....193

3 Exemplo #3 – Defasada em até 2 ciclos com predominância de ausência de observações 197

Referências Bibliográficas 199

Lista de figuras

Figura 1 – Arquitetura funcional geral de gerenciamento. 5

Figura 2 – Exemplo de sintomas observados por um sistema de diagnóstico. 9

Figura 3 – Exemplo de sintomas decorrentes de uma anomalia. 10

Figura 4 – Arquitetura de um sistema de gerenciamento 19

Figura 5 – DBM visto como a interação de observações e predições, extraído de (DAVIS, 1988) 21

Figura 6 –Exemplo de conhecimento núcleo e interpretativo, extraído de Abu-Hanna (1994) 26

Figura 7 – Principais classes de modelos, segundo Abu-Hanna (1994) 26

Figura 8 – Representação gráfica do sistema BOX. 28

Figura 9 – Exemplo de modelo estrutural para o sistema BOX. 28

Figura 10 – Exemplo de modelo comportamental para o sistema BOX. 29

Figura 11 - Exemplo de modelo comportamental corrigido para o sistema BOX 29

Figura 12 – Exemplo de modelo comportamental interpretativo para o sistema BOX 30

Figura 13 – Exemplo de modelo funcional interpretativo para o sistema BOX. 30

Figura 14 - Exemplo de modelo causal interpretativo para o sistema BOX 31

Figura 15 – Arquitetura geral do sistema SMARTS 36

Figura 16 – Exemplo de descrição utilizando a linguagem MODEL. 38

Figura 17 – Exemplo de grafo de correlação, derivado do exemplo de (KLIGER, 1995) 39

Figura 18 – Exemplo de matriz de correlação derivada do grafo de correlação 40

Figura 19 – Vetores código da matriz de correlação da Figura 18 41

Figura 20 – Exemplo de vetor código derivado de uma observação 41

Figura 21 – Exemplo de livro-código de raio 0,5 (distância mínima=1), extraída de (KLIGER, 1995) 42

Figura 22 – Distância entre códigos para o exemplo da Figura 18 42

Figura 23 – Distância entre códigos para o exemplo da Figura 21 42

Figura 24 – Exemplo de livro-código de raio 1,5 (distância mínima=3), extraída de Klinger (1995) 43

Figura 25 – Distância entre códigos para o exemplo da Figura 24 43

Figura 26 – Sistema de diagnóstico atemporal sobre único instante. 46

Figura 27 – Sistema de diagnóstico atemporal sobre uma coleção de sintomas. 47

Figura 28 – Sistema de diagnóstico atemporal sobre múltiplos instantes. 48

Figura 29 – Sistema de diagnóstico temporal 49

Figura 30 – Sistema de diagnóstico temporal variante no tempo. 50

Figura 31 – Observações em um sistema de diagnóstico 53

Figura 32 – Exemplo de valores de objetos gerenciados SNMP da MIB-2 que não possuem significado isoladamente 53

Figura 33 – Exemplo de mapeamento da observação em estados. 54

Figura 34 - Exemplo de mapeamento da observação em transição de estados 56

Figura 35 – Exemplo de uma classe de alarmes gerados por uma plataforma de gerenciamento. 56

Figura 36 – Exemplo de diagrama de transição de estados. 56

Figura 37 – O papel do agente de gerenciamento. 63

Figura 38 – Exemplo de valor de objeto gerenciado sem significado isoladamente. 66

Figura 39 – Exemplo de objeto gerenciado composto no tempo. 66

Figura 40 – Visão geral da arquitetura de um sistema de diagnóstico tradicional. 67

Figura 41 - Papel do gerente na obtenção de estados (valores) dos objetos intermediários... 67

Figura 42 – Exemplo de objeto gerenciado e seu valor 68

Figura 43 – Exemplo de representação intermediária e seu valor 68

Figura 44 – Exemplo do objeto da diferença do estado observado no objeto intermediário em relação ao objeto gerenciado. 69

Figura 45 – Observações em um sistema de diagnóstico. 70

Figura 46 – Arquitetura tradicional de infra-estrutura para um sistema de diagnóstico 70

Figura 47 – Arquitetura da infra-estrutura para diagnóstico com o modelador de observações. 71

Figura 48 – Exemplo de relacionamento das definições apresentadas. 73

Figura 49 – Exemplos de relações “necessariamente causa” e “possivelmente causa”. 74

Figura 50 - As diferentes combinações dos intervalos de tempo entre causa e efeito. 75

Figura 51 – Exemplo de relações causais de início e término retardado. 75

Figura 52 – Exemplo de representação alternativa para as relações da Figura 51. 76

Figura 53 – Exemplo de relação causal entre anomalias 76

Figura 54 – Exemplo de relações causais entre anomalias. 77

Figura 55 – Exemplo de relação causal entre anomalias 77

Figura 56 – Exemplo de grafo de relação causal entre anomalias 77

Figura 57 – Exemplos de relacionamento causal entre anomalia e sintoma. 78

Figura 58 – Exemplo de relações causais entre anomalia e sintoma. 78

Figura 59 – Exemplo de anomalias sem sintomas diretamente associados. 79

Figura 60 – Exemplo de anomalias sem sintomas diretamente associados. 79

Figura 61 – Diferentes anomalias causando o mesmo sintoma. 80

Figura 62 – Diagrama causal entre anomalias e sintomas. 80

Figura 63 – Relação “causa” entre sintomas. 81

Figura 64 – Exemplo de grafo causal, adaptado de Kinger (1995) 81

Figura 65 – Exemplo de observação não defasada. 86

Figura 66 – Exemplo de expressão de uma observação defasada em até 1 ciclo. 86

Figura 67 - Exemplo de observação defasada em até 1 ciclo 87

Figura 68 – Exemplo de expressão de uma observação defasada em até 2 ciclos 87

Figura 69 - Exemplo de observação defasada em até 2 ciclos que apresenta defasagem de até 1 ciclo 88

Figura 70 – Exemplo de uma observação defasada em até 2 ciclos. 89

Figura 71 – Exemplo de intervalos no qual não existe informação atualizada sobre o objeto gerenciado 90

Figura 72 – Exemplo de relação causal. 91

Figura 73 – Exemplo de observações que utilizam períodos diferentes de amostragem 92

Figura 74 – Exemplo do impacto da defasagem de observações no relacionamento causal devido à utilização de diferentes períodos de amostragem..... 92

Figura 75 - Exemplo de observações que utilizam fases diferentes de amostragem. 93

Figura 76 - Exemplo do impacto da defasagem de observações no relacionamento causal devido à utilização de diferentes fases de amostragem 93

Figura 77 – Exemplo de modelamento da observação do estado E de um objeto gerenciado95

Figura 78 – Exemplo de modelamento da observação de todos os estados de um objeto gerenciado..... 95

Figura 79 – Modelo tradicional de monitoramento utilizado por um sistema de diagnóstico. 97

Figura 80 – Entidades envolvidas no modelo tradicional de geração de observações para um sistema de diagnóstico. 97

Figura 81 – Novo modelo de monitoramento utilizado por um sistema de diagnóstico. 98

Figura 82 - Entidades envolvidas no novo modelo de geração de observações para um sistema de diagnóstico. 98

Figura 83 – Exemplo de modelamento de uma observação não defasada. 99

Figura 84 - Exemplo de modelamento de uma observação defasada em até 1 ciclo. 100

Figura 85 - Exemplo de modelamento de uma observação defasada em até 2 ciclos 101

Figura 86 - Exemplo de modelamento de uma observação defasada em até 1 ciclo, com a completude de seus estados. 101

Figura 87 - Exemplo de modelamento de uma observação, defasada em até 2 ciclos, com a completude de seus estados. 102

Figura 88 – Exemplo de intervalo de incerteza em uma observação não defasada..... 103

Figura 89 – Exemplo de intervalo de incerteza gerado devido a falta de uma amostragem em observações defasadas em até 1 ciclo. 104

Figura 90 - Exemplo de intervalo de incerteza gerado devido à falta de uma amostragem em observações defasadas em até 2 ciclos. 104

Figura 91 – Exemplo de intervalo de incerteza em uma observação defasada em até 1 ciclo, no instante T_{c1} 106

Figura 92 - Exemplo de intervalo de incerteza em uma observação defasada em até 1 ciclo, no instante $T_{c2} = T_{c1} + P$ 106

Figura 93 - Exemplo de intervalo de incerteza em uma observação defasada em até 2 ciclos, no instante T_{c1} 107

Figura 94 - Exemplo de intervalo de incerteza em uma observação defasada em até 2 ciclos, no instante $T_{c2} = T_{c1} + P$ 107

Figura 95 – Exemplo de intervalo de incerteza existente no início do processo de monitoração para observações defasadas em até 1 ciclo. 108

Figura 96 - Exemplo de intervalo de incerteza existente no início do processo de monitoração para observações defasadas em até 1 ciclo. 109

Figura 97 – Exemplo de modelamento de atraso para observações não defasadas. 110

Figura 98 – Exemplo de modelamento de atraso para observações defasadas em até 1 ciclo. 110

Figura 99 - Exemplo de modelamento de atraso para observações defasadas em até 2 ciclos. 110

Figura 100 – Posicionamento do Modelador na arquitetura do sistema 111

Figura 101 – Principais constantes e variáveis utilizadas nos algoritmos. 111

Figura 102 – Algoritmo de modelamento de observações imediatas. 112

Figura 103 – Algoritmo de modelamento para observações defasadas em até 1 ciclo. 115

Figura 104 – Algoritmo de modelamento para observações defasadas em até 2 ciclos. 119

Figura 105 – Exemplos de *clusters* em observação defasada em até 1 ciclo. 120

Figura 106 – Exemplos de *clusters* em observação defasada em até 2 ciclos. 120

Figura 107 – Uma das possíveis formas de realizar a normalização. 122

Figura 108 – Relação causal entre uma anomalia e um sintoma. 123

Figura 109 - Relação causal entre uma anomalia e dois sintoma. 124

Figura 110 – Relação causal entre duas anomalias. 125

Figura 111 – Relações causais entre anomalias 125

Figura 112 – Interface do método de diagnóstico 126

Figura 113 – Modelo de diagnóstico temporal utilizado no sistema SiDiR-t 127

Figura 114 – Modelos utilizados no diagnóstico..... 128

Figura 115 – Exemplo de modelo de configuração do domínio físico..... 128

Figura 116 – Exemplo de modelo de configuração de domínio de subrede..... 129

Figura 117 – Exemplo de grafo causal derivado dos modelos..... 129

Figura 118 – Exemplo de grafo causal dos sintomas e anomalias de um servidor 130

Figura 119 – Relação de sintomas e anomalias apresentadas no grafo causal da Figura 118. 131

Figura 120 – A tarefa de diagnóstico segundo Benjamins (1993). 132

Figura 121 – Métodos propostos por Benjamins (1993) para a tarefa “detecção de sintomas” 133

Figura 122 – Papel das entidades de apoio no sistema de diagnóstico..... 134

Figura 123 - Métodos utilizado pelo sistema SiDiR-t para a tarefa de detecção de sintomas. 134

Figura 124 - Métodos propostos por Benjamins (1993) para a tarefa de geração de hipóteses. 136

Figura 125 – Métodos de geração de hipóteses utilizado pelo sistema SiDiR-t..... 136

Figura 126 – Visão funcional geral de um sistema de gerenciamento 145

Figura 127. Modelo geral de gerenciamento..... 148

Figura 128 – Formato da mensagem SNMP 150

Figura 129 – Principais formas de interação entre gerente e agente. 150

Figura 130. Componentes de um agente DMI (GHETIE, 1998) 152

Figura 131. Interação entre gerentes, agentes e objetos gerenciados 153

Figura 132. *Common Management Service Element* 154

Figura 133. Pilha de protocolos no gerenciamento OSI (GHETIE, 1998). 155

Figura 134. Exemplo da definição de uma classe de objeto..... 156

Figura 135. Exemplo de parte de uma árvore de herança. 157

Figura 136. Árvore de registro. 158

Figura 137. Exemplo de cada um dos quatro tipos possíveis de definição de escopo. 159

Figura 138. Exemplo de uma rede de telecomunicação 160

Figura 139. Relacionamento da TMN com a rede de telecomunicações 162

Figura 140. Inter-relacionamento entre os blocos funcionais da TMN 163

Figura 141. Camadas funcionais de suporte ao gerenciamento 164

Figura 142. Um elemento de uma rede de telecomunicações 165

Figura 143. Planos de gerenciamento e as camadas funcionais 166

Figura 144 – Ilustração das relações “causa” e “efeito de”. 167

Figura 145 – Exemplo de grafo causal, adaptado de (KLIGER, 1995). 167

Figura 146 – Exemplo de grafo causal com a eliminação de ciclos. 168

Figura 147 – Exemplo de relação causal representada na forma de conjunto matemático. . 169

Figura 148 – Exemplo de relação causal representada graficamente na forma de grafo, adaptado de (KLIGER, 1995) 169

Figura 149 – Exemplo de relação causal representada na forma de tabela. 170

Figura 150 – Grafo de correlação derivado do grafo causal. 170

Figura 151 – Exemplo de modelo determinístico causal. 172

Figura 152 – Exemplo de modelo probabilístico causal. 172

Figura 153 – Exemplo de modelo temporal causal. 172

Figura 154 – Código de bloco binário 173

Figura 155 – (a) Regra de Hamming; (b) Palavra de código de Hamming 175

Figura 156 – Exemplo de representação de conhecimento temporal através de grafos. 184

Lista de tabelas

Tabela 1 – Principais diferenças entre o gerenciamento de rede de telecomunicações e do gerenciamento de rede de dados. 8

Tabela 2 – Alguns tipos de operações de correlação, extraído de JAKOBSON (1999) 17

Tabela 3 – Ações a serem tomadas de acordo com a seqüência de estados obtidos na amostragem. 113

Tabela 4 - Ações a serem tomadas de acordo com a seqüência de estados obtidos na amostragem. 116

Tabela 5 – Alguns serviços oferecidos por uma rede de telecomunicação. 161

Tabela 6 – Blocos funcionais da TMN. 163

Tabela 7 – Tabela de transitividade de relações, extraída de (ALLEN, 1993). 185

Lista de Definições

Definição 1: Diagnóstico baseado em consistência. 33

Definição 2: Diagnóstico abdutivo. 33

Definição 3: Componente 61

Definição 4: Anomalia 62

Definição 5: Classe de Objeto Gerenciado (COG). 63

Definição 6: Objeto Gerenciado (OG) 64

Definição 7: Valor de objeto gerenciado 65

Definição 8: COG composta no tempo 65

Definição 9: Classe de objeto intermediário (COI) 68

Definição 10: Objeto Intermediário (OI) 68

Definição 11: Valor de objeto intermediário 69

Definição 12: COI composta no tempo 69

Definição 13: Observação 70

Definição 14: Sintoma 71

Definição 15: IO(E) - Intervalo de ocorrência de estado 95

Definição 16: IC(E) - Intervalo de certeza de ocorrência de estado 96

Definição 17: IPI(E) - Intervalo de possibilidade de início de ocorrência de estado 96

Definição 18: IPT(E) - Intervalo de possibilidade de término de ocorrência de um estado 96

Definição 19: Aglomerado (*cluster*) de intervalos de observação 119

Definição 20: Cluster formato normal 121

Definição 21: Intersecção entre dois *clusters* na forma normal 122

Definição 22: Relação causal. 168

Definição 23: Semi-anel. 171

Definição 24: Ponto de tempo 178

Definição 25: Relações binárias entre pontos de tempo 178

Definição 26: Intervalo de tempo convexo. 180

Definição 27: Relações de ordem binária sobre intervalos convexos. 180

Definição 28: Outras relações 181

Definição 29: Duração de um intervalo convexo. 182

Definição 30: Interseção de intervalos convexos. 182

Definição 31: Cobertura de intervalos convexos 183

Definição 32: Relações de ordem binária entre pontos de tempo e intervalos convexos. 186

Lista de Abreviaturas

API	*Application Program Interface*
ASN.1	*Abstract Syntax Notation . 1*
ACSE	*Association Control Service Element*
CCITT	*Consultative Committee for International Telegraph and Telephone*
COG	Classe de Objeto Gerenciado
COI	Classe de Objeto Intermediário
CMIP	*Common Management Information Protocol*
CMIS	*Common Management Information Service*
CMISE	*Common Management Information Service Element*
COSS	*Common Object Services Specification*
CPU	*Central Processor Unit*
DBM	Diagnóstico baseado em modelo
DC	Diagnóstico contínuo
DMC	Diagnóstico de momento corrente
DMI	*Desktop Management Interface*
DMP	Diagnóstico de momento passado
DMTF	*Desktop Management Task Force*
DNS	*Domain Name System*
DN	*Distinguished Name*
FDN	*Full Distinguished Name*
GDMO	*Guidelines for the Definition of Management Objects*
GSM	*Global System for Mobile Communications*

IC	Intervalo de Certeza
IETF	*Internet Engineering Task Force*
II	Intervalo de Incerteza
IIMC	*ISO and Internet Management Coexistence*
IP	*Internet Protocol*
IPI	Intervalo de Possibilidade de Início
IPT	Intervalo de Possibilidade de Término
ISO	*International Organization for Standardization*
ITU-T	*International Telecommunications Union, Telecommunications Standard Section*
IDL	*Interface Definition Language*
JIDM	*Joint Inter-Domain Management Group*
MF	*Mediation Function*
MD	*Mediation Device*
MIB	*Management Information Base*
MIS-User	*Management Information Service – User*
NMF	*Network Management Forum*
MOC	*Managed Object Class*
MO	*Managed Object*
NAND	*Not And*
OI	Objeto Intermediário
OG	Objeto Gerenciado
OS	*Operations Systems*
OSF	*Operations System Function*
OSI	*Open System Interconnection*

QA	*Q Adaptor*
QAF	*Q Adaptor Function*
RDN	*Relative Distinguished Name*
ROSE	*Remote Operation Service Element*
SDBM	Sistema de Diagnóstico Baseado em Modelo
SMI	*Structure of Management Information*
SMASE	*Systems Management Application Service Element*
SMAE	*Systems Management Application Entity*
TCP	*Transmission Control Protocol*
TMN	*Telecommunications Management Network*
TMF	*Tele Management Forum*
TINA	*Telecommunications Information Network Architecture*
TINA-C	*Telecommunications Information Network Architecture Consortium*
UDP	*User Datagram Protocol*
VLAN	*Virtual Local Area Network*
WS	*Workstation*
WSF	*Workstation Function*

1. Introdução

Cada vez mais as corporações dependem do ambiente computacional para apoiar suas atividades diárias. Algumas corporações são totalmente dependentes de seu ambiente computacional fazendo com que a ocorrência de determinadas falhas gerem prejuízos imensos.

Por outro lado, o ambiente computacional está a cada dia mais complexo. Isto é devido principalmente à utilização de diversas tecnologias de sistemas e de comunicação interoperáveis aliada à complexidade cada vez maior destes sistemas e destes protocolos de comunicação.

Por esse motivo a tarefa de diagnóstico de falhas em um ambiente computacional torna-se cada vez mais complexa e difícil. São vários os problemas. Um dos principais é manter atualizada uma documentação a respeito da topologia e configurações utilizadas neste ambiente, fundamental durante um processo de detecção de falhas. Para isto, sistemas automáticos ou semi-automáticos de descoberta de topologia e configuração têm sido implementados. Porém, a tarefa de geração automática da topologia e configuração do ambiente é dificultada principalmente pela utilização de protocolos com configuração dinâmica (por exemplo, roteamento dinâmico), introdução de novos protocolos (por exemplo, VLAN e protocolos de qualidade de serviço) e novas funcionalidades no ambiente de comunicação (por exemplo, filtragem de pacotes em equipamentos de interconexão).

Porém, mais complexa ainda tem se tornado a tarefa do operador para a descoberta de falhas. Na atualidade, é praticamente impossível a um operador acompanhar o comportamento de um ambiente computacional, mesmo com o apoio de plataformas de gerenciamento. Isto é devido principalmente à quantidade de informações que necessitam ser relacionadas (cujas observações necessitam ser definidas previamente) em intervalos de tempo muito pequenos e ao entendimento do significado de informação.

Os sistemas de diagnóstico, portanto, apesar de complexos, podem ser extremamente úteis no gerenciamento da infra-estrutura computacional.

1.1 Detecção de problemas

A detecção de problemas em um ambiente computacional é uma tarefa necessária em diversos ambientes corporativos. Os principais desafios existentes na condução de um processo de detecção de problemas são:

- **Diversidade de classes de equipamentos e de sistemas**: Atualmente em um ambiente computacional existe uma diversidade muito grande de classes de equipamentos e de sistemas: diferentes tipos de servidores, de roteadores, de chaveadores (*switches*), de *no-breaks*, de sistemas operacionais, de serviços de rede e de aplicações;
- **Quantidade de equipamentos**;
- **Quantidade de informações disponibilizadas**: Os agentes de monitoração de rede de dados (tipicamente agentes SNMP) disponibilizam uma quantidade de informações muito grande. Por exemplo, um agente SNMP de um único equipamento disponibiliza milhares de objetos gerenciados somente relacionados à MIB-2. Isto sem contar as outras MIBs que o agente suporta;
- **Qualidade das informações disponibilizadas**: Grande parte das informações disponibilizadas por sistemas de gerenciamento não são efetivamente úteis na detecção de falhas ou não são úteis isoladamente;
- **Dificuldade de entendimento das informações pelos operadores**: Somente operadores muito experientes têm conhecimento efetivo do significado de cada informação disponibilizada por sistemas de gerenciamento;
- **Dificuldade de correlacionamento das informações**: A tarefa de correlacionamento das informações disponibilizadas por sistemas de gerenciamento é complexa e mesmo impossível de ser realizada manualmente;
- **Configuração dos equipamentos**: A tarefa de detecção de problemas necessita do conhecimento da topologia e configuração atual do ambiente computacional. Algumas plataformas de gerenciamento possuem módulos

que permitem esta funcionalidade. Porém, nem todas as informações necessárias são obtidas e disponibilizadas.

Em uma rede de telecomunicação, por exemplo, as anormalidades que ocorrem durante a operação da rede provocam a emissão automática de notificações (eventos) que são direcionadas ao centro de gerência de rede. A partir das notificações recebidas, o operador humano deve tentar identificar a anomalia ocorrida. Alguns centros de gerência podem chegar a receber dezenas de milhares de eventos diariamente, tornando cada vez mais complexo o processamento manual destes eventos. Diversos fatores contribuem para esta situação:

- Um equipamento pode gerar diversos eventos em decorrência de uma única falha;
- A falha pode ser intrinsecamente intermitente, o que causa o envio de uma notificação a cada ocorrência;
- A falha de um componente pode resultar no envio de um evento toda vez que o serviço prestado por este componente é invocado;
- Uma única falha pode ser detectada por múltiplos componentes da rede, cada um deles emitindo um evento;
- A falha de um dado componente pode afetar diversos outros componentes, causando a propagação da falha.

Mesmo sistemas automáticos de correlação ou diagnóstico devem conviver com situações que dificultam o processamento como:

- **Defasagem da observação**: A observação realizada por um sistema de diagnóstico a respeito do ambiente computacional pode se apresentar defasada em relação à ocorrência efetiva;
- **Perda de observações:** É possível que ocorra perda de observação, ou seja, perda de notificação ou requisição de informação de estado de um objeto;
- **Existência de ruídos.** As observações a respeito do ambiente computacional representam estados aproximados. Pode ocorrer ruído, ou seja, uma observação errada a respeito do estado do ambiente computacional.

1.1.1 Sistemas de apoio

A tarefa de detecção de anomalias em um sistema distribuído fica a cargo de uma equipe de operadores geralmente fazendo parte da equipe de gerência de redes e sistemas ou integrada a esta. Para auxílio nesta tarefa existem alguns sistemas de apoio:

- console de alarmes (disponível nas plataformas de gerenciamento);
- sistema de correlação de alarmes (que pode ser agregado às plataformas de gerenciamento);
- sistema de diagnóstico.

Existe ainda outro sistema de apoio denominado “reparo” que permite, após identificada a anomalia, corrigir ou contornar (ativando planos de contingência) o problema. Este trabalho não abordará os sistemas de reparo.

1.1.2 Gerenciamento

O gerenciamento de um sistema distribuído não é tarefa trivial. Primeiramente, por envolver diferentes classes de sistemas, sejam *hardware* ou *software* (sistema operacional, aplicações), em geral de diferentes fabricantes. Em segundo lugar, cada sistema possui uma função de atuação no ambiente, implicando na necessidade de entendimento de seu papel no ambiente e de que forma seu comportamento pode ser observado. Por último, porque cada sistema pode possuir uma sintaxe ou API específica para seu gerenciamento.

Para facilitar, foram criados padrões de gerenciamento, os quais possuem como objetivo principal possibilitar a interoperabilidade entre sistemas no que se refere ao gerenciamento. Um padrão de gerenciamento define, entre diversos outros aspectos:

- o modelo de dados;
- o protocolo de comunicação ou API (no caso de acesso local).

Gerenciar uma determinada entidade envolve duas atividades principais: monitoração (somente observação de seu estado) e controle (alteração de seu estado).

O inter-relacionamento destas tarefas com os subsistemas de apoio está ilustrado na Figura 1.

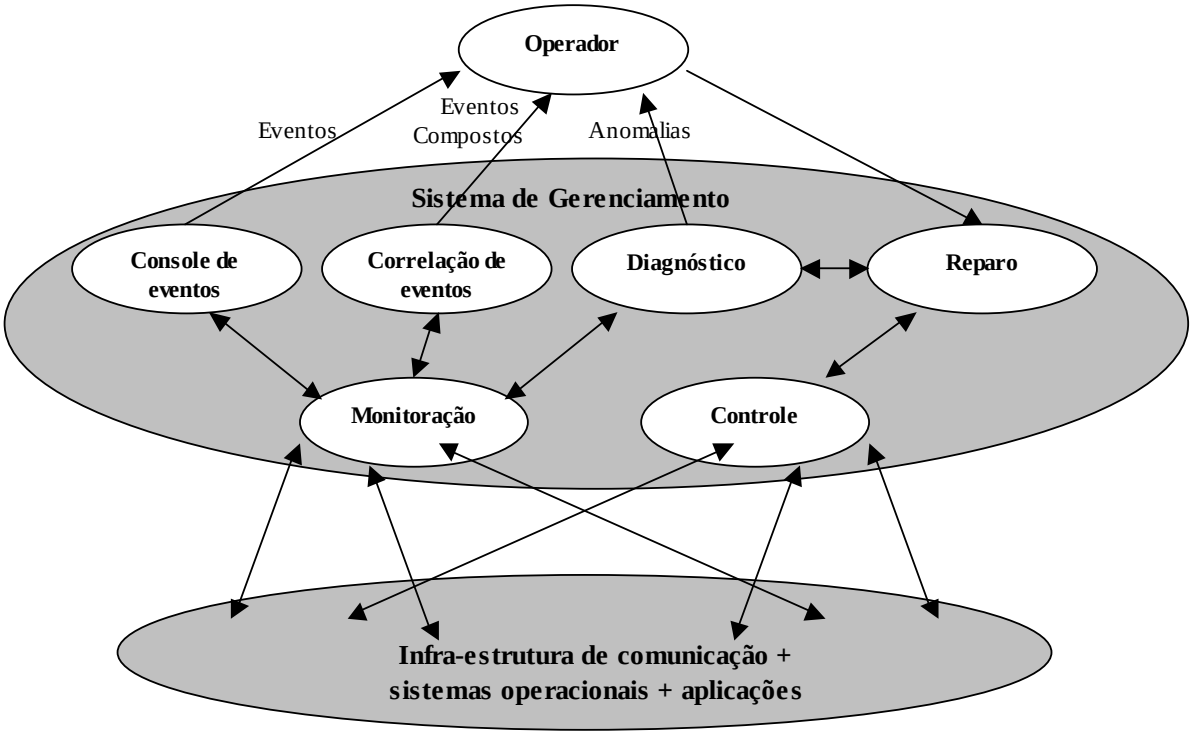


Figura 1 – Arquitetura funcional geral de gerenciamento.

Os principais protolocos de gerenciamento padronizados são[1]:

- SNMP (*Simple Network Management Protocol*), protocolo padrão de gerenciamento para Internet definido pelo IETF (*Internet Engineering Task Force*);
- Padrão de gerenciamento OSI (*Open System Interconnection*) definido pela ISO (*International Organization for Stardardization*);
- DMI (*Desktop Management Interface*);
- TMN (*Telecommunications Management Network*), padrão de gerenciamento, baseado no padrão OSI, definido pela ISO para um ambiente de telecomunicação.

[1] O Anexo 1apresenta uma breve descrição a respeito dos protocolos de gerenciamento.

1.1.3 Console de alarmes das plataformas de gerenciamento

Os consoles de alarmes são subsistemas tradicionais em plataformas de gerenciamento. Concentram todos os alarmes recebidos em um banco de dados interno e apresentam uma listagem, geralmente em ordem cronológica, em uma interface gráfica possibilitando ao operador navegar e selecionar alarmes. O operador pode selecionar um alarme e obter maiores informações a respeito do alarme. Além disso, pode adicionalmente disponibilizar as seguintes funcionalidades:

- suprimir um alarme em função de sua prioridade;
- selecionar um conjunto de alarmes em função de alguma característica;
- compressão de alarmes idênticos consecutivos em um único alarme.

Também é possível ao operador eliminar os alarmes já observados da lista apresentada.

1.1.4 Sistemas de correlação de alarmes

Segundo Jakobson (1993 e 1999) a **correlação de evento** é um procedimento que tem por objetivo fornecer uma interpretação conceitual de um determinado grupo de eventos de forma a atribuir um novo significado a esse grupo de eventos.

A correlação de eventos geralmente tem por objetivo reduzir a quantidade de notificações de eventos transferidos aos operadores do sistema de gerência de rede, aumentando o conteúdo semântico das notificações resultantes.

A correlação de eventos é extensivamente utilizada nas redes de telecomunicações. Ela pode ser aplicada a qualquer uma das cinco áreas funcionais de gerência definidas pelo ITU-T. Contudo, segundo Meira (1997b), a maioria das aplicações encontradas na literatura está relacionada à gerência de falhas, que é a mais elementar e mais importante.

1.1.5 Sistemas de diagnóstico

Diferentemente de um sistema de correlação de eventos, um **sistema de diagnóstico** de anomalias[2] tem por objetivo a identificação das causas raiz a partir de um conjunto de sintomas que estejam sendo observados no ambiente gerenciado. O sistema de diagnóstico pode utilizar, eventualmente, uma técnica de correlação de eventos. Porém, antes de se chegar à causa raiz, pode ser necessário a formulação de um conjunto de hipóteses, as quais precisarão ser validadas.

O desafio de identificar a causa raiz reside nos seguintes fatos:

- um simples problema pode gerar diversos sintomas, alguns dos quais propagados e distantes da causa raiz;
- a causa raiz compartilha de diversos sintomas com outras possíveis causas. A causa raiz pode não ser óbvia analisando os sintomas individualmente;
- a causa raiz do problema pode não ser observável. Por exemplo, pode não ser possível monitorar a planta de energia de um determinado andar. Mas, se for detectado que os equipamentos de um andar não estão operantes (sintomas), pode-se supor a ocorrência de uma falha de energia (diagnóstico);
- uma vez que a causa raiz tenha sido identificada, todos os sintomas recebidos causados por ela são explicados e não necessitam de análise posterior;
- é possível utilizar este fato (anomalia raiz) para explicar outros sintomas que venham a ocorrer, desde que possam ter sido causados por esta.

É desejável que um sistema de diagnóstico de falhas para um ambiente distribuído possua um modelo da configuração do ambiente, processe o fluxo de observações em tempo real e seja capaz de trabalhar com dados incompletos.

Um sistema de diagnóstico também deve informar, se possível, sobre o impacto causado por determinada anomalia no sistema. Isto é importante, pois o operador

---

[2] Denominado na literatura de sistema de diagnóstico de falhas. O termo anomalia será preferencialmente utilizado neste trabalho. A seção 5.1.2 apresenta a definição adotada para anomalia.

seria somente informado a respeito da anomalia raiz e nunca de suas conseqüências, isto é, das outras anomalias por ela causadas.

1.2 Motivação

Apesar de se mostrarem eficientes em redes de telecomunicações, os sistemas de correlação não se mostram tão eficientes no gerenciamento de uma rede de dados. Nas redes de telecomunicações é utilizado extensivamente o modelo de gerenciamento OSI (BRISA, 1993), no qual os agentes possuem grande autonomia e flexibilidade de ativação de funções gerenciais. O comportamento do ambiente é observado pelo gerente através, principalmente, de notificações. Além disso, o gerenciamento é majoritariamente *out-of-band*, ou seja, não utiliza o meio gerenciado para tráfego de informações gerenciais. Assim, o problema de perda de informações gerenciais é minimizado.

	Rede de Telecomunicações	**Redes de Dados**
Protocolo mais utilizado	CMIP	SNMP
Agente de monitoração	complexo	simples
Meio de transmissão utilizado no gerenciamento	*out-of-band e in-band*	*in-band*
Principal método de obtenção de observações	notificação	amostragem periódica (*polling*)
Observação típica	não defasada	defasada em até 1 ciclo defasada em até 2 ciclos
Perda de observações	raro	freqüente

Tabela 1 – Principais diferenças entre o gerenciamento de rede de telecomunicações e do gerenciamento de rede de dados.

Nas redes de dados, pelo contrário, é utilizado extensivamente o gerenciamento SNMP (RFC1155, RFC1157, RFC1212, RFC1212, RFC1213, RFC1214, RFC1215). Os agentes são mais simples, obrigando o gerente a utilizar largamente a técnica de amostragem periódica (*polling*) para a coleta de informações gerenciais. Como o

gerenciamento é *in-band* (utiliza o próprio meio gerenciado para a transmissão de informações gerenciais), a possibilidade de perda de informações gerenciais aumenta consideravelmente. Aliado a isto, o protocolo da camada de transporte utilizado, o UDP, fornece um serviço datagrama não confiável. O gerenciamento *in-band* também pode impor restrições a respeito da quantidade de informações gerenciais observadas. Por este motivo, um sistema de diagnóstico para redes de dados deve se concentrar nos objetos gerenciados mais importantes. Somente quando necessário devem ser realizadas observações adicionais. Uma comparação das principais diferenças entre o gerenciamento de uma rede de telecomunicação e uma rede de dados é apresentada à Tabela 1.

A tarefa de diagnóstico pode também apresentar outros desafios, como mostrado no exemplo da Figura 2. Uma alta taxa de ocupação de CPU poderia explicar o fato de um determinado serviço de rede, por exemplo um servidor DNS, não responder por um determinado período. Porém, da forma como estas observações estão representadas no tempo, as técnicas convencionais iriam indicar que não existe relacionamento entre estes sintomas.

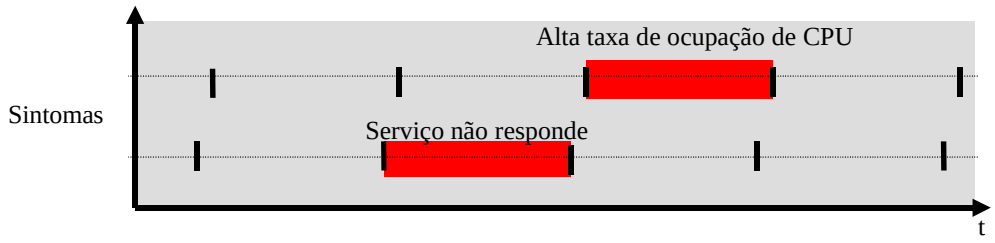


Figura 2 – Exemplo de sintomas observados por um sistema de diagnóstico.

Apesar da interseção dos intervalos das observações ser vazia, indicando sintomas não correlacionados, ainda assim o sintoma “serviço não responde” pode ser causado pela anomalia “alta taxa de ocupação de CPU”. Muitas vezes as observações são apresentadas defasadas no tempo, dificultando sua correlação ou diagnóstico. A Figura 3 mostra como esses sintomas podem efetivamente estar associados a uma mesma anomalia.

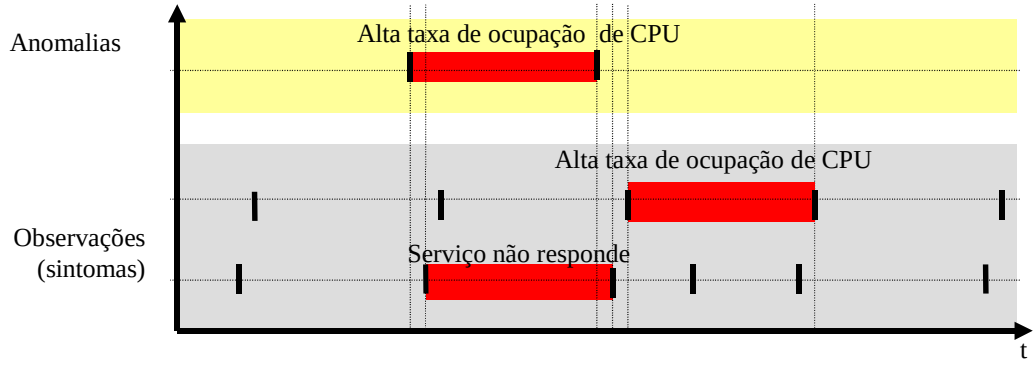


Figura 3 – Exemplo de sintomas decorrentes de uma anomalia.

Neste exemplo, “Serviço não responde” é um sintoma que pode estar defasado em até 1 ciclo de amostragem enquanto que “Alta taxa de ocupação de CPU” é um sintoma que pode estar defasado em até 2 ciclos (vide capítulo 6).

Um sistema de diagnóstico que se utiliza das observações (sintomas) presentes em um ponto no tempo (diagnóstico atemporal baseado em único instante[3]) não seria capaz de relacionar esses sintomas a uma única causa, gerando um “falso negativo”. Da mesma forma, podem existir “falsos positivos” (por exemplo, em sistemas de diagnóstico atemporal sobre uma coleção de sintomas ou diagnóstico atemporal sobre múltiplos instantes), ou seja, ser informada uma causa raiz a partir de um conjunto de sintomas não relacionados efetivamente.

Um dos objetivos deste trabalho é modelar a observação de forma a acrescentar “intervalos de possibilidade” e “intervalos de certeza” em relação à ocorrência da anomalia. O problema do recebimento das observações de forma periódica impõe ao sistema de diagnóstico um “período de incerteza” a respeito do intervalo de ocorrência da anomalia. O problema de defasagem no tempo é menos crítico em um ambiente de gerenciamento OSI pois os intervalos de amostragem tendem a ser menores. Isto porque o controle da amostragem é realizado diretamente pelo agente e são utilizadas notificações quando da mudança de estado.

O problema da falta de observação também é abordado neste trabalho. A inexistência de um sintoma (efetivamente passível de ser observado) não necessariamente indica que não exista uma anomalia. Por algum motivo pode não ser possível observar o

[3] Descrito no capítulo 3.

estado de uma entidade em um determinado momento. Devido ao gerenciamento "*in-band*" é possível ocorrer:

- Impossibilidade de obtenção de novas observações;
- Perda de notificações.

Por este motivo é importante também modelar os intervalos de tempo no qual ocorra impossibilidade de realização de observações de um determinado comportamento do ambiente.

1.3 Objetivo

Esta tese tem como objetivo abordar alguns aspectos relevantes dos sistemas de diagnóstico baseado em modelo (DAVIS, 1984; CONSOLE, 1990; ABU-HANNA, 1990; BENJAMINS, 1993), particularmente aqueles dedicados ao diagnóstico temporal de sistemas distribuídos, a saber: (1) proposição de um modelamento temporal das observações utilizadas por um sistema de diagnóstico que trate a imprecisão temporal; (2) proposição de um modelamento das observações utilizadas que trate a ausência de informações gerenciais (ausência de observações) e (3) mostrar como um método de diagnóstico pode utilizar informações sobre imprecisão temporal para possibilitar um resultado de diagnóstico mais preciso.

1.4 Justificativa

Existem diversas arquiteturas utilizadas em um sistema de diagnóstico. Neste trabalho será utilizada a baseada em modelos. A grande vantagem de um sistema de diagnóstico baseado em modelos é a possibilidade elaboração de métodos de diagnóstico reusáveis para uma determinada classe de modelos (ABU-HANNA, 1993; LEMOS, 1997; LEMOS, 1998). Para determinadas tarefas de diagnóstico, muitas vezes é necessário desenvolver um método específico. Este é o caso do trabalho descrito em (BERNAL, 1999b; FONTANINI, 2002) no qual foi desenvolvido um método específico para a identificação de falhas em uma rede de dados que utiliza um modelo de configuração do ambiente. Mesmo sendo um método

específico para um determinado tipo de problema, pode ser utilizado em diferentes ambientes, bastando criar o modelo de configuração para cada um.

Este trabalho propõe a utilização de um "Sistema de Coleta de Informações Gerenciais" no qual o sistema de diagnóstico tenha um maior controle do processo de monitoração de forma a:

- possibilitar que as observações repassadas pelo sistema de coleta ao sistema de diagnóstico incluam as incertezas temporais e os intervalos de impossibilidade de monitoração;
- possibilitar ao sistema de diagnóstico a programação dos objetos a serem monitorados e, quando relevante, a programação do período de amostragem;
- possibilitar a descoberta da configuração do ambiente.

Outro aspecto importante desta tese é a caracterização de um sistema de diagnóstico em relação a sua interação com o ambiente[4] e publicado em (BERNAL, 1999b). A forma de interação com o ambiente influi nas técnicas de diagnóstico utilizadas, daí a importância de sua caracterização.

1.5 Trabalhos Relacionados

Relacionado diretamente à tese, existe o seguinte trabalho interno:

- SiDiR: Este projeto foi iniciado em 1996 com o objetivo de desenvolver um sistema baseado em modelos para a área de diagnóstico de falhas em redes de computadores. A primeira versão do sistema (BARROS, 1999; BERNAL, 1999b; LEMOS, 1999; WAINER, 2000) incluía a utilização de uma plataforma de gerenciamento OSI como interface com os objetos gerenciados (rede). Incluía também um sistema de descoberta de configuração da rede e operava por transição de estados. A segunda geração, desenvolvida a partir de 1999 já não utilizava uma plataforma de gerenciamento, possuindo total controle sobre a monitoração. Em outro trabalho, Fontanini (2002) propôs um

[4] Apresentada no capítulo 4.

método de diagnóstico de falhas de comunicação e de desempenho que otimiza o processo de monitoração e intervalo de detecção de sintomas através do controle da observação pelo sistema de diagnóstico. A eficiência é em relação ao intervalo entre testes de comunicação, principalmente nos equipamentos de interconexão, e também em relação à diminuição da utilização do meio com informações de gerenciamento.

Alguns trabalhos relacionados ao diagnóstico e correlacionamento são:

- Sistema de diagnóstico para a plataforma SIS (BARBOSA, 2002). É um sistema de diagnóstico baseado em modelo estrutural e funcional para a área de gerenciamento de redes de telecomunicações (TMN);
- SMC – *Search with Model-based Constraints* (ZHENG, 2002): Apresenta um algoritmo de correlação de eventos para uma rede de telecomunicação GSM baseado no modelo de configuração apresentado em (BARROS, 1999). Classifica os alarmes a serem correlacionados em “escopo-correlacinado”, “intra-correlacionado” e “inter-correlacionado”, conforme se originem de um mesmo equipamento, de uma mesma classe de equipamentos (neste caso, da mesma camada de gerenciamento) ou de diferentes classes de equipamentos;
- SMART (KLIGER, 1995): Utiliza uma técnica de correlacionamento de eventos. Um modelo de eventos representa as informações dobre vários eventos e seu relacionamento causal. Utiliza um modelo de configuração para gerar o modelo de eventos. A correlação é baseada na técnica de livro-código (*codebook).*

Existem atualmente alguns sistemas comerciais de diagnóstico de falhas para rede local (BOARDMAN 2002):

- Smarts InCharge Solutions Suíte (KLIGER, 1995);
- Netcool/Omnibus (Micromuse);
- Spectrum xsight (Apisma Management Technologies);
- Managed Objects Formula e Business Service Analyse;
- Patrol Enterprise Manager (BMC Software).

Destes, o produto “Smarts InCharge Solutions” apresenta diversas características interessantes. É um sistema de diagnóstico baseado na correlação de eventos que utiliza a técnica de livro-código (*codebook*). Esta técnica patenteada possibilita uma correlação muito eficiente. Porém, é susceptível à perda de amostragens e defasagens de tempo nas observações. É baseada em modelo, possuindo um sistema gerador de modelo e gerador do livro-código[5].

1.6 Estrutura da tese

A tese é composta por 8 capítulos. O capítulo 2 descreve os sistemas de apoio à gerencia de falhas: console de eventos, sistemas de correlação e sistemas de diagnóstico. Apresenta também uma descrição a respeito do sistema de correlação/diagnóstico SMARTS que se utiliza da técnica de livro-código para otimizar o processo de correlação. Em seguida, no capítulo 3 são discutidos os sistemas de diagnóstico com dimensão temporal. O capítulo 4 descreve algumas formas possíveis de classificação dos sistemas de diagnóstico: quanto ao tipo da observação recebida; quanto ao controle do processo de observação; e quanto ao momento de diagnóstico. Uma definição a respeito de termos utilizados no diagnóstico é apresentada no capítulo 5 que também apresenta a caracterização da relação causal e as formas de relacionamento entre anomalia e sintoma. No capítulo 6 é realizada uma caracterização do processo de observação, mostrando as entidades envolvidas e os problemas apresentado na monitoração de ambiente distribuído em redes de dados. Neste capítulo é também apresentado uma proposta de modelamento da observação através da utilização de intervalos de possibilidade, certeza e incerteza. O capítulo 7 descreve um método de diagnóstico temporal e, por último, a conclusão no capítulo 8.

Também existem 5 anexos: o primeiro descreve os sistemas de gerenciamento. O anexo 2 descreve as relações causais, sendo seguido pela descrição dos códigos de Hamming, no anexo 3. O anexo 4 aborda o tema de representação do tempo e,

[5] A seção 2.6 apresenta mais detalhes a respeito do sistema SMARTS e da técnica de livro-código.

finalmente, o anexo 5 mostra exemplos de como os intervalos de tempo são construídos.

2. Sistemas de Correlação e Diagnóstico

Este capítulo descreve os sistemas de correlação e diagnóstico. Ao final é descrito um dos principais sistemas de diagnóstico para ambiente distribuído chamado “SMARTS”, baseado em modelo que utiliza a técnica de correlação por livro-código.

2.1 Sistemas de correlação de eventos

Segundo Jakobson (1993 e 1999) a **correlação de eventos** é um procedimento que tem por objetivo fornecer uma interpretação conceitual a respeito de determinado grupo de eventos de forma a atribuir um novo significado a eles.

A correlação de eventos geralmente tem por objetivo reduzir a quantidade de notificações de eventos transferidos aos operadores do sistema de gerência de rede, aumentando o conteúdo semântico das notificações resultantes.

Sistemas de correlação de eventos são extensivamente utilizados para a identificação de problemas em redes de telecomunicações. Existem vários motivos para isso. Primeiramente, o tratamento manual dos eventos tem se mostrado inviável devido à complexidade e extensão de uma rede de telecomunicações. Segundo, o modelo de gerenciamento OSI fornece ao agente um grau de funcionalidade muito grande, possibilitando a programação de geração de eventos (alarmes).

A correlação de eventos pode ser aplicada a qualquer uma das cinco áreas funcionais de gerência definidas pelo ITU-T. Contudo, segundo Meira (1997b), a maioria das aplicações encontradas na literatura está relacionada à gerência de falhas, que é a mais elementar e mais importante.

A Tabela 2 mostra alguns tipos de operações de correlação, descritas em (JAKOBSON, 1999).

Tipo	**Exemplo**	**Obs.**
Compressão	[a, a, ..., a] → a	
Filtragem	[a, p(a) < H] → ∅	
Supressão	[a, C] → ∅	
Contagem	[n x a] → b	
Intensificação	[n x a, p(a)] → a' , p(a') > p(a)	P(x) = prioridade de x = minor, critical, major
Generalização	[a, a subconjunto de b] → b	
Especialização	[a, a superconjunto de b] → b	
Relação temporal	[a T b] → c	
Clustering	[a, b, ... T, and, or, not] → c	Relações complexas

Tabela 2 – Alguns tipos de operações de correlação, extraído de JAKOBSON (1999)

A tarefa de correlação de eventos é um processo dinâmico que utiliza um fluxo de eventos de entrada, a topologia do ambiente e outras informações. É também um processo dependente do tempo.

Para Meira (1997a e 1997b) e Jakobson (1999), diversos métodos podem ser utilizados para realizar o correlacionamento de eventos, dentre os quais:

- filtragem simples;
- baseado em regras;
- baseado em casos;
- baseado em modelo;
- máquina de estados finitos;
- redes neurais;
- lógica difusa;
- livro código (*codebook*);
- teste de equipamento (correlação ativa)

2.2 Sistemas de diagnóstico

Diferentemente de um sistema de correlação, cujo objetivo é "compactar" a informação a respeito de um conjunto de eventos recebidos para um operador, um sistema de diagnóstico tem por objetivo encontrar a causa raiz que explique um determinado conjunto de sintomas observados no ambiente.

É desejável que um sistema de diagnóstico apresente as seguintes características:

- rapidez do resultado do diagnóstico;
- tratamento de múltiplas anomalias simultâneas;
- tolerância à defasagem de fase e período de amostragem[6];
- tolerância à observações defasadas (em até 1 ou até 2 ciclos)[7];
- tolerância à perda de observações[8];
- apresentação de diagnóstico provisório, mesmo com informações parciais;
- detecção de anomalias intermitentes;
- esperteza na manipulação de efeitos não imediatos;
- esperteza na manipulação de anomalias que somente algumas vezes são causadas por outra;
- suporte a diagnóstico hierárquico;
- alterações de topologia ou configuração do ambiente computacional não devem causar alterações no sistema de raciocínio do sistema de diagnóstico;
- informação sobre o impacto que a anomalia detectada pode causar no ambiente.

---

[6] O problema de defasagem de fase e período de amostragem é descrito na seção 6.2.3.

[7] O problema de defasagem das observações é descrito na seção 6.1.

[8] O problema da perda de observações é descrito na seção 6.5.1.

Dentre elas existem características que são mutuamente exclusivas, e estão citadas exatamente para possibilitar a comparação entre diversos sistemas.

2.3 Sistemas de diagnóstico

Diagnóstico, segundo Benjamins (1993) em seu trabalho a respeito de métodos de resolução de problemas, é "*a tarefa de identificar a causa relacionada a anomalias que se manifestam através de alguns comportamentos observáveis*".

Existe uma outra tarefa relacionada, porém distinta, que é o **Reparo**, tarefa associada à correção ou contingenciamento de uma anomalia. A Figura 4 ilustra estas tarefas em um sistema de gerenciamento de redes.

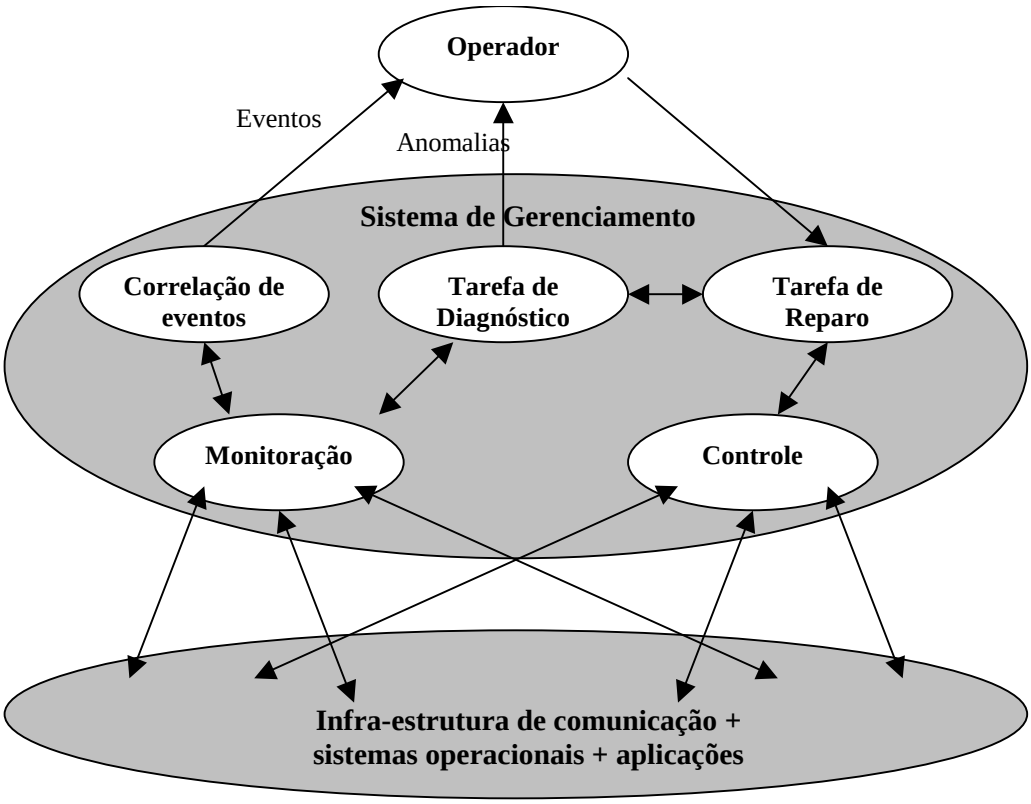


Figura 4 – Arquitetura de um sistema de gerenciamento

2.4 Classes de sistemas de diagnóstico

Na literatura existem inúmeros exemplos de tipos de sistemas de diagnóstico. Estes podem ser divididos, segundo Benjamins (1993), em duas classes principais:

- baseados em heurística;
- baseados em modelo.

2.4.1 Sistemas de diagnóstico baseados em heurística

Esta classe de sistema utiliza conhecimento que é baseado na experiência com o dispositivo/sistema em consideração. Tipicamente, este conhecimento é obtido por meio de entrevistas com um especialista ou através de casos ocorridos no passado. Geralmente, o conhecimento experimental consiste em regras (heurísticas) que associam sintomas às possíveis causas.

Segundo Benjamins (1993), as principais limitações associadas aos sistemas baseados em heurística são:

- **Erros em casos de pequenas divergência**: se o problema diferir ligeiramente do anotado pelo especialista ou pelos casos passados o sistema abruptamente é incapaz de realizar a inferência;
- **Necessidade de experiência passada antes da concepção do sistema de diagnóstico**: aquisição de conhecimento depende da existência de experiência humana ou coleta de casos passados. Isto implica em que experiência deveria estar disponível, antes de implantar o sistema;
- **Explicação limitada de como a solução foi alcançada**: desde que o conhecimento em tais sistemas é geralmente representado na forma de regras do tipo condição-ação, as explicações consistem no alinhavo das regras que foram usadas no processo de inferência. Tal alinhavo pode ser de difícil compreensão para o entendimento humano, dificultando a aceitação de tais soluções.

- **Dificuldade de reuso**: uma vez que um sistema baseado em heurística tenha sido construído para um sistema específico, é difícil o reuso de partes em outros sistema.

2.4.2 Sistemas de diagnóstico baseados em modelo

O paradigma básico de Diagnóstico Baseado em Modelo (DBM) pode ser entendido pelo confronto de observações e predições (DAVIS, 1988), como ilustrado na Figura 5.

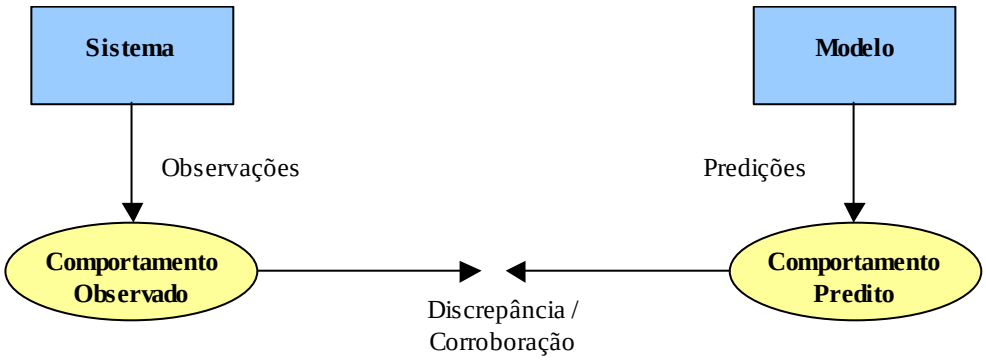


Figura 5 – DBM visto como a interação de observações e predições, extraído de (DAVIS, 1988)

Geralmente, os sistemas de diagnóstico são aplicados a dispositivos específicos. Nestes casos, o modelo descreve tipicamente os componentes do dispositivo, suas conexões e o comportamento de cada componente. Geralmente, este modelo descreve o **comportamento correto** do sistema. Porém, pode representar também o **comportamento anômalo** do sistema.

Um sistema de diagnóstico baseado em modelo assume que o modelo não contenha erros. Porém, segundo (DAVIS, 1988), dificilmente um modelo não conterá erros sobretudo devido às aproximações utilizadas. Cada modelo pode conter várias simplificações e pressupostos que não necessariamente representam com exatidão o sistema.

Um sistema de diagnóstico baseado em modelo, apesar disso, oferece algumas vantagens em relação às limitações apresentadas por um sistema de diagnóstico baseado em heurística:

- **Comportamento mais robusto**: É capaz de manipular problemas rotineiros assim como novos e inesperados. Um sistema baseado em heurística é capaz de manipular somente problemas précodificados. Um sistema baseado em modelo atua sobre uma informação explícita (o modelo do dispositivo) que, a princípio, deveria explicar todas as possíveis anomalias;
- **Facilidade de aquisição de conhecimento.** O modelo do sistema pode geralmente ser extraído do projeto do sistema e tipicamente descreve a estrutura (componentes e conexões) e comportamento (relacionamento entrada-saída) dos componentes;
- **Melhor explicação a respeito da anomalia**: A explicação a respeito da anomalia pode ser construída a partir das associações realizadas sobre o modelo;
- **Melhor reusabilidade**: Um modelo definido para um componente pode ser reutilizado no modelamento do mesmo componente presente em um outro sistema.

Porém, ainda existem diversas dificuldades na concepção de sistemas baseados em modelo:

- **Problema de modelamento**: Refere-se à dificuldade de construção do modelo do dispositivo ou sistema (DAVIS, 1988). No caso específico do diagnóstico de ambiente computacional é inviável e desnecessário construir um modelo preciso;
- **Processamento computacional**: Custo computacional de processamento do modelo frente aos diferentes sintomas e hipóteses. Geralmente, o diagnóstico baseado em modelo é intratável computacionalmente. Para minimizar tal problema é necessário o acréscimo de heurísticas para guiar o processo computacional (BENJAMINS, 1993).

2.5 Sistemas de Diagnóstico Baseados em Modelo

Em seu trabalho a respeito de métodos de resolução de problemas, Benjamins (1993) cita dois importantes componentes presentes em um sistema de diagnóstico baseado em modelo (SDBM): o modelo do sistema e o processo de diagnóstico. Console (1998-b), além destes apresenta também o tipo de dado temporal como um componente importante. Porém, a forma com que a observação incorpora informações temporais é somente uma parte do problema. Esta tese estende esta definição, englobando todo o modelamento da observação, chamando este componente de modelo da observação. Assim, é possível citar como sendo os principais componentes de um SDBM:

- **Modelo do sistema**: Modelo de representação que descreve o sistema. Um sistema pode ser descrito por diversas maneiras;
- **Processo de diagnóstico do problema**: Para cada modelo utilizado (ou conjunto de modelos) podem existir diferentes processos de diagnóstico, que utilizam tais modelos de diferentes maneiras;
- **Modelo da observação**: Trata do modelamento da observação a ser utilizada pelo sistema de diagnóstico. A observação necessita ser coletada e tratada de forma a torná-la usável pelo sistema de diagnóstico. O processo de observação pode ser complexo e sujeito a imprecisões de diversas ordens (por exemplo: imprecisão temporal e de valor). O modelo da observação tem por objetivo descrever a observação e, eventualmente, suas imprecisões ou aproximações.

2.5.1 Classificação de sistemas de diagnóstico

Existem diversas dimensões ortogonais nas quais um sistema de diagnóstico pode ser classificado:

- quanto ao tipo do modelo utilizado;
- quanto ao tipo de modelo de funcionamento (correto ou anômalo);
- quanto a ser quantitativo ou qualitativo;

- quanto ao tipo de inferência (abdutivo[9] ou baseado em consistência);
- Quanto a forma de modelamento temporal.

2.5.2 Classificação quanto aos modelos utilizados

Existem diversas formas de descrever um sistema para uso em diagnóstico. Segundo Abu-Hanna (1994), existem duas classes primitivas:

- **Independente de contexto (modelos núcleo)**, modelos que descrevem o comportamento dos componentes (entidades primitivas) que constituem o sistema, independentemente da maneira como elas foram combinadas entre si;
- **Dependente de contexto (modelos interpretativos),** modelos que descrevem os componentes do sistema, utilizando um nível maior de abstração, uma interpretação do sistema por um determinado ponto de vista, um determinado contexto. Geralmente não possui uma representação completa do comportamento do componente. A representação geralmente é direcionada de forma a facilitar um processo de diagnóstico.

Os modelos núcleo possuem a visão do mundo mais próxima da realidade e de forma independente da tarefa a ser executada no domínio. Os modelos interpretativos são mais abstratos e focalizam, normalmente, a realização de uma tarefa específica sobre o dispositivo, sendo o modelo uma interpretação do comportamento do sistema de acordo com algum critério.

Os modelos independentes de contexto modelam o sistema através da descrição de cada um de seus componentes individualmente. Isto, na maior parte dos caso, só é possível se for limitado o "mundo" a ser modelado. Segundo Abu-Hanna (1994), um modelo é independente de contexto quando atender aos seguintes critérios:

- **Princípio da localidade**: Implica que a descrição de cada componente pode somente se referir a parâmetros internos do componente ou interfaces com o exterior;

---

[9] Do inglês *abductive*.

- **Compatibilidade ontológica entre os componentes**: Significa que a descrição dos componentes utiliza os mesmos termos para descrever componentes potencialmente relacionados;
- **Princípio da "ausência de função na estrutura"[10]**: Significa que a descrição de um componente não inclui nenhuma funcionalidade que dependa de alguma configuração específica do sistema (neste último caso, a análise do componente dependeria de um contexto). Assim, é possível descrever o comportamento de todo o sistema como sendo a soma do comportamento de seus componentes individuais. Ou seja, cada componente pode ser descrito individualmente e localmente, independente do contexto no qual opera. Isto implica também na facilidade de reuso do modelo em outros sistemas;
- **Princípio de mundo fechado sobre o comportamento do componente**: A descrição do modelo do componente é completa com respeito à visão do sistema no mundo real. Ou seja, não existem influências comportamentais que não estejam modeladas.

Um modelo dependente de contexto é menos geral que um modelo independente de contexto. Um modelo dependente de contexto geralmente é utilizado para:

- complementar o modelo independente de contexto;
- aumentar a eficiência do processo de raciocínio através de uma descrição em mais alto nível.

Como um modelo interpretativo pode ser considerado como uma interpretação de um modelo núcleo, geralmente existe uma relação entre eles. Também é verdade que um modelo dependente de contexto em um nível de abstração pode ser independente de contexto em outro nível. A Figura 6 mostra um exemplo que ilustra esta situação.

[10] Do inglês "*no function in structure*".

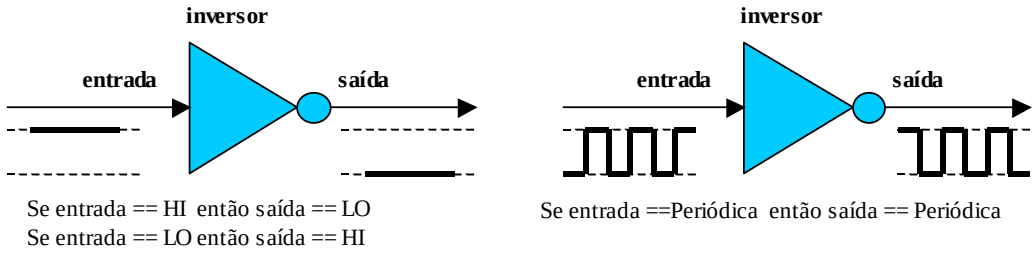


Figura 6 –Exemplo de conhecimento núcleo e interpretativo, extraído de Abu-Hanna (1994)

Em Abu-Hanna (1994) são relacionados os principais modelos que podem ser derivados a partir destas duas classes de modelos (núcleo e interpretativo). A Figura 7 mostra uma classificação parcial. A classificação independente de contexto inclui os modelos topológico, estrutural e comportamental. Como a classificação em dependente de contexto ou independente de contexto é relativa à visão do mundo, o ponto de referência utilizado formam as primitivas do modelo estrutural.

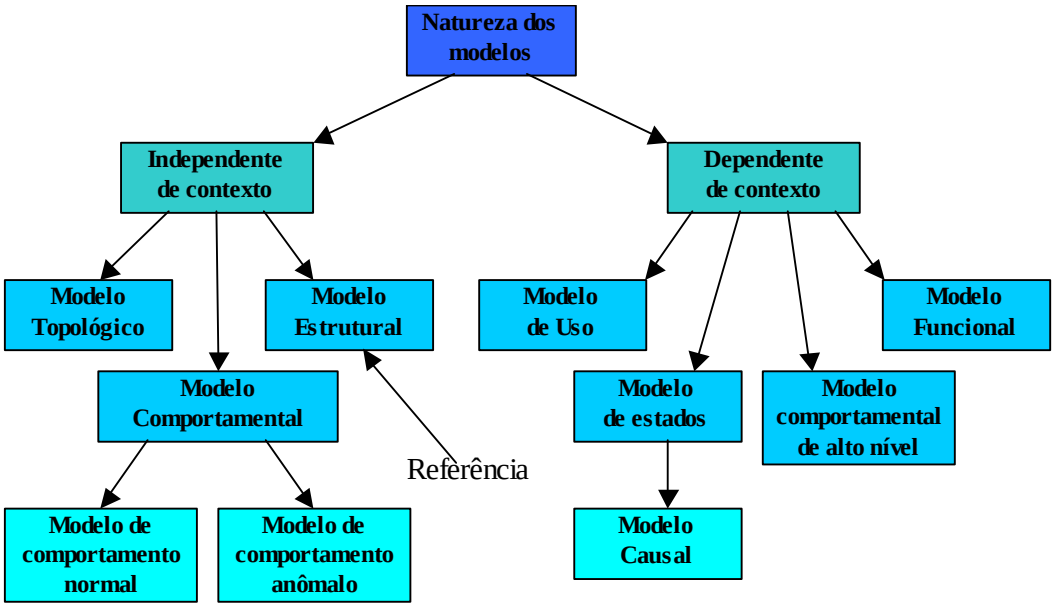


Figura 7 – Principais classes de modelos, segundo Abu-Hanna (1994)

Modelo estrutural: Descreve a estrutura do sistema, seus componentes básicos e as interconexões entre tais componentes;

Modelo comportamental: Descreve o comportamento de cada componente do sistema através de seus estados internos ou valores apresentados em suas interfaces externas, geralmente relacionando entrada e saída. Segundo Benjamins (1993),

existem dois tipos de relações entrada-saída: regras de simulação e regras de inferência (também chamadas de regras *forward* e *backward*). As regras de simulação têm o papel de predizer a saída dada uma certa entrada. Assim, descrevem o comportamento supondo um estado normal dos componentes. As regras de inferência têm o papel de descrever conclusões válidas a respeito do comportamento de um sistema. Elas computam a entrada de um componente em função de suas saídas;

Modelo topológico: Descreve a topologia utilizada em sistemas que são compostos por subsistemas fracamente acoplados;

Modelo funcional: Um modelo funcional descreve o dispositivo através das funções e subfunções que o sistema é capaz de realizar. Funções são abstrações da estrutura e comportamento. Estes modelos interpretam o conhecimento comportamental de acordo com funções pretendidas pelo projetista. Tais modelos contém: funções, parâmetros de função e restrições entre parâmetros e equações;

Modelo de estados: Representa os estados comportamentais relevantes do sistema, além de relações entre tais estados. As relações normalmente refletem causalidade (relação causal);

Modelo de uso: Reflete a percepção do dispositivo pela visão do usuário. É importante para mapear os termos utilizados pelos usuários fim aos termos utilizados em outros modelos do sistema.

Para exemplificar o uso de alguns destes modelos, pode-se considerar o exemplo mostrado na Figura 8, extraído de Abu-Hanna (1994), que consiste em um sistema chamado BOX composto por dois componentes eletrônicos, um inversor e um dispositivo lógico NAND.

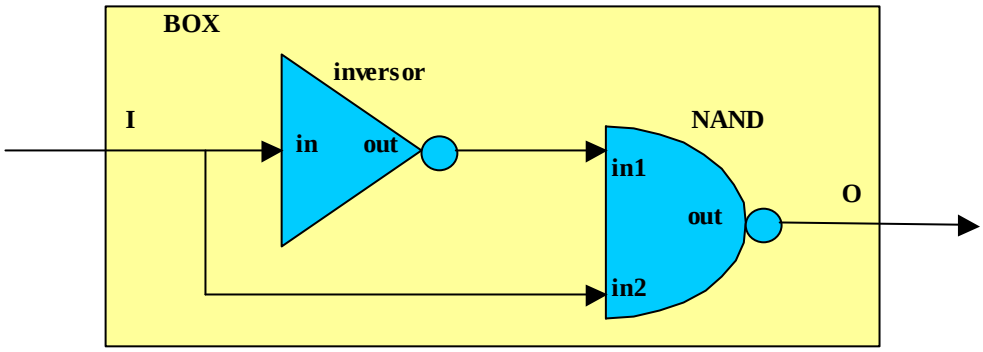


Figura 8 – Representação gráfica do sistema BOX.

A Figura 9 mostra um exemplo de modelo estrutural que representa o sistema BOX.

Inversor
componente(inversor)
porta(inversor,in,(input))
porta(inversor,out,(output))

NAND
componente(nand)
porta(nand,in1,(input))
porta(nand,in2,(input))
porta(nand,out,(output))

BOX
componente(box)
porta(box,I,(input))
porta(box,O,(output))
composto(Box,inversor,nand)

Interconexões
conectado(Box,I,inversor,in)
conectado(Box,I,nand,in2)
conectado(inversor,out,nand,in1)
conectado(nand,out,Box,O)

Figura 9 – Exemplo de modelo estrutural para o sistema BOX.

A Figura 10 mostra um possível modelo comportamental para o sistema BOX. Utilizando os modelos estrutural e comportamental é possível simular seu comportamento. A simulação indica que a saída “O” é sempre “HI” independente da entrada aplicada. Entretanto, esta previsão não representa o comportamento observado do sistema BOX no mundo real. No mundo real, devido à latência de propagação de sinal imposta pelo inversor, um pulso “LO” é produzido na saída “O”.

Inversor

output = out
inputs = [in]
outEQN(out):
  if (in == HI)
  then out = LO
  else out = HI

NAND

output = out
inputs = [in1, in2]
outEQN(out):
  if (in1 == HI) and (in2 == HI)
  then out = LO
  else out = HI

Figura 10 – Exemplo de modelo comportamental para o sistema BOX.

O problema citado anteriormente surgiu devido à não inclusão, no modelo de sistema, da latência de propagação de sinal imposta pelos componentes primitivos. Isto viola a propriedade “princípio de mundo fechado sobre o comportamento do componente”.

Um engenheiro de conhecimento poderia criar um modelo interpretativo que capturasse o comportamento do sistema BOX através da observação de como o sistema se comporta no mundo real. Entretanto (a) resultaria em perda de generalidade, uma das características importantes oferecidas pelos modelos núcleo e (b) necessitaria de que o modelo interpretativo fosse construído empiricamente. Neste caso é preferível corrigir o modelo comportamental, como mostrado na Figura 11.

Inversor

output = out
inputs = [in]
delay_time = 0.5
outEQN(out):
  if (in == HI) at t0
  then out = LO at t0 + delaytime
  else out = HI at t0 + delaytime

NAND

output = out
inputs = [in1, in2]
delay_time = 0,2
outEQN(out):
  if (in1 == HI) and (in2 == HI) at t0
  then out = LO at t0 + delaytime
  else out = HI at t0 + delaytime

Figura 11 - Exemplo de modelo comportamental corrigido para o sistema BOX

Existem também várias outras formas de representar o sistema BOX. Devido ao modo como os componentes do sistema BOX foram conectados e à natureza de seu comportamento, foram observados padrões especiais de comportamento que podem

ser representados em um modelo. Particularmente, existe o seguinte relacionamento entrada(“I”)/saída(“O”): quando ocorre uma transição de “LO” para “HI” na entrada “I”, ocorre um pulso na saída “O”.

Assim, é possível conceber um modelo interpretativo alternativo para o sistema BOX, mostrado na Figura 12.

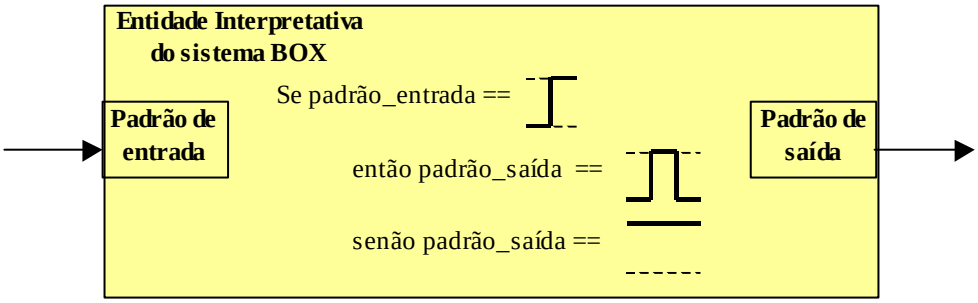


Figura 12 – Exemplo de modelo comportamental interpretativo para o sistema BOX

Se a geração do pulso for intencionada pelo projetista (sendo sua ausência uma anomalia) é possível criar um modelo funcional interpretativo, mostrado na Figura 13.

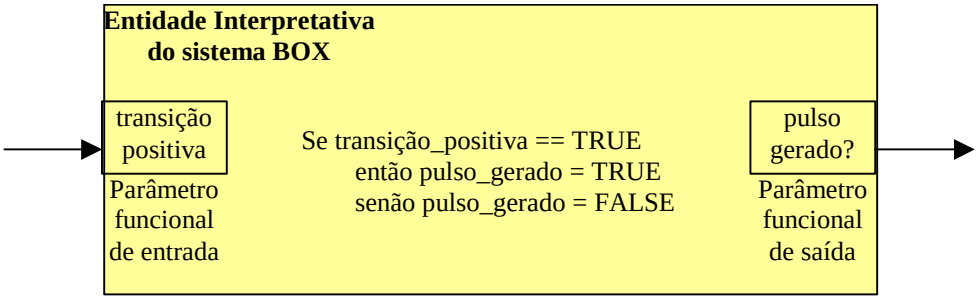


Figura 13 – Exemplo de modelo funcional interpretativo para o sistema BOX

É possível ainda representar o dispositivo através de um modelo causal, como mostrado na Figura 14.

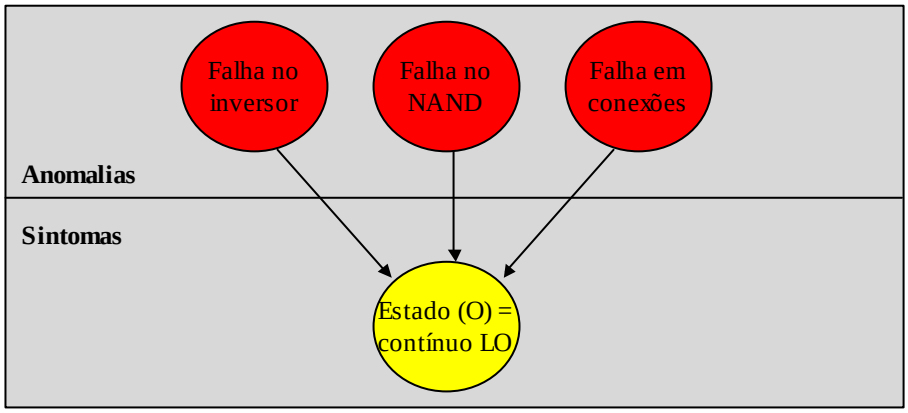


Figura 14 - Exemplo de modelo causal interpretativo para o sistema BOX

2.5.2.1 A utilização de múltiplos modelos

Segundo Lemos (1997), a decisão sobre quais tipos de modelos devem ser disponibilizados para a realização da tarefa de diagnóstico deve levar em conta vários aspectos:

- A complexidade do domínio e conseqüentemente a complexidade de construção e manipulação do modelo. O modelo núcleo comportamental, por exemplo, é geralmente adequado para dispositivos muito simples. Para domínios complexos é mais adequado modelar em um nível maior de abstração de forma que os aspectos relevantes à tarefa sejam focalizados;
- Robustez e precisão do diagnóstico. Alguns modelos descrevem aspectos complementares (estrutural e comportamental, estrutural e causal). Por exemplo, observado um determinado sintoma no sistema, o raciocínio pode ser iniciado com o auxílio de um modelo causal para encontrar os componentes suspeitos (causas) e, em um próximo passo, os componentes suspeitos podem ser investigados individualmente fazendo uso do modelo estrutural e comportamental.

2.5.3 Classificação quanto ao tipo de modelo de funcionamento

Outra classificação que pode ser aplicada a um sistema de diagnóstico é quanto ao tipo de modelo de funcionamento: correto ou anômalo.

O **modelo de funcionamento correto** expressa como o sistema normalmente funciona enquanto que o **modelo de funcionamento anômalo** expressa como o sistema se comporta quando certas anomalias ocorrem.

Quando o **modelo de funcionamento correto** é utilizado, uma anomalia é caracterizada pela ausência do comportamento correto. A grande vantagem é que as anomalias não precisam ser modeladas de antemão e que o modelo, teoricamente, poderia ser concebido a partir de informações de seu projeto, o que facilita o processo de aquisição de conhecimento. Nenhuma informação a respeito de possíveis anomalias precisa ser representada e, por este motivo, o modelo não é dependente de nenhum processo de experiência. Porém, existem algumas limitações: algumas anomalias não podem ser detectadas: aquelas que expressam comportamentos ortogonais às funcionalidades projetadas. Por exemplo, um circuito eletrônico pode produzir fumaça. Entretanto, no projeto (e conseqüentemente no modelo) não existe qualquer funcionalidade associada à produção de fumaça.

O **modelo de funcionamento anômalo** torna explícita as anomalias de um sistema. Tais sistemas podem fornecer explicações mais específicas a respeito do que está errado no sistema. Entretanto, o raciocínio somente com modelos de funcionamento anômalo requerem a completude do modelo se for desejado ter certeza de uma solução correta. Tal completude é difícil de atingir.

Modelo de funcionamento correto não é factível em sistemas complexos, como um sistema distribuído, devido à impossibilidade de construção de modelos núcleo para tais sistemas.

2.5.4 Classificação quando a ser quantitativo ou qualitativo

Um sistema de diagnóstico pode também ser classificado como quantitativo ou qualitativo. Modelos quantitativos são em geral descritos matematicamente, sendo mais complexos, porém mais precisos que modelos qualitativos. Modelos qualitativos descrevem o sistema através de termos qualitativos. Isto pode facilitar o processo de modelamento, tornando factível o modelamento de sistemas complexos.

2.5.5 Classificação quanto ao tipo de inferência

Outra classificação de sistemas de diagnóstico se refere à forma de inferência utilizada. Existem dois tipos de inferência que podem ser realizados por um sistema de diagnóstico: abdutivo[11] ou baseado em consistência.

Definição 1: Diagnóstico baseado em consistência.

Seja DESCR a descrição do sistema (modelo), COMPS o conjunto de componentes do sistema, OBS o conjunto de observações e Δ um subconjunto de componentes do sistema ($\Delta \subseteq$ COMPS). Um **diagnóstico baseado em consistência** para um problema de diagnóstico (DESCR, COMPS, OBS) é definido como o conjunto mínimo de componentes Δ tal que:

$$DESCR \cup OBS \cup \{ Anômalo(c) \mid c \in \Delta\} \cup \{ \neg Anômalo(c) \mid c \in COMPS - \Delta\}$$

seja consistente.

Nesta definição, Δ é o resultado do diagnóstico e representa o conjunto de componentes que apresentam anomalias.

Definição 2: Diagnóstico abdutivo.

Seja DESCR a descrição do sistema (modelo), COMPS o conjunto de componentes do sistema, OBS o conjunto de observações e Δ o conjunto de modos (estados) associados a cada componente do sistema ($\Delta = \{m(c) \mid m \in Modos(c), \forall c \in COMPS\}$). Um **diagnóstico abdutivo** para um problema de diagnóstico (DESCR, COMPS, OBS) é definido como o conjunto Δ de suposição de modos de comportamento tal que:

$DESCR \cup \Delta \models OBS$ e [12]

$DESCR \cup \Delta \cup OBS$ é consistente

[11] Do inglês "*abductive*".

[12] Notação: se x |= y então a conclusão y é conseqüência sintática da premissa x.

Nesta definição, Δ é o resultado do diagnóstico e representa o conjunto de estados supostos para cada componente do sistema, incluindo portanto estados normais (componente normal) e estados anômalos (dos componentes que apresentam problemas).

Para exemplificar a diferença, considere-se o exemplo apresentado por Benjamins (1993), no qual a solução "tanque vazio" é uma solução para o problema em um sistema de diagnóstico baseado em consistência de um carro que apresenta os seguintes sintomas "não dá partida" e "luzes não acendem".

Já um sistema de diagnóstico abdutivo, "tanque vazio" não poderia ser uma solução para o problema, pois, apesar de consistente, não cobre a observação "luzes não acendem". Neste caso, "bateria falha" poderia ser uma solução válida.

Como a cobertura é uma condição mais forte que a consistência, o diagnóstico abdutivo é mais restritivo que o diagnóstico baseado em consistência no sentido de que a abdução necessita de mais suposições para ser verdadeira. Em particular, a completude do modelo de dispositivo é assumida.

Existem ainda **alternativas intermediárias**. Uma delas seria considerar a cobertura de observações anômalas e ser consistente com as observações normais. Neste caso, "tanque vazio" poderia ser uma solução válida na ocorrência das seguintes observações: "motor não liga" (observação anômala) e "luzes acendem" (observação normal).

2.5.6 Classificação quanto a forma de modelamento temporal

Os sistemas de diagnóstico também podem ser classificados quanto à forma de modelamento temporal. Esta classificação está detalhada no capítulo 3 desta tese.

2.6 Estudo de caso: O sistema SMARTS

O sistema SMARTS (KLIGER, 1995; OHIE, 1997[a]; OSHIE, 1997b; BROADMAN, 2002; SMARTS, 2000; WHITE, 1998) é um sistema de diagnóstico baseado em modelo que se utiliza da técnica de livro-código (*codebook*). Apesar de esta ser de

correlação, nada impede que seja utilizada por um sistema de diagnóstico. Quando associada com um modelo do sistema permite a descoberta das causas raiz das anomalias.

A Figura 15 mostra uma visão geral da arquitetura do sistema. Os monitores são os responsáveis pela interação com o ambiente a fim de detectar as anomalias e conseqüente geração de alarmes que são correlacionados (KLIGER, 1995) pelo módulo de correlação, utilizando a técnica de livro-código (*codebook*). O ambiente é modelado utilizando a linguagem MODEL (OSHIE, 1997a; OSHIE 1998b) que permite descrever a forma como as anomalias são propagadas através dos componentes do sistema e como as entidades se relacionam. O sistema de descoberta gera informações sobre a topologia e configuração do ambiente que é utilizado pelo módulo de gerador de livros código, juntamente com o modelo do sistema, para gerar os livros código necessários.

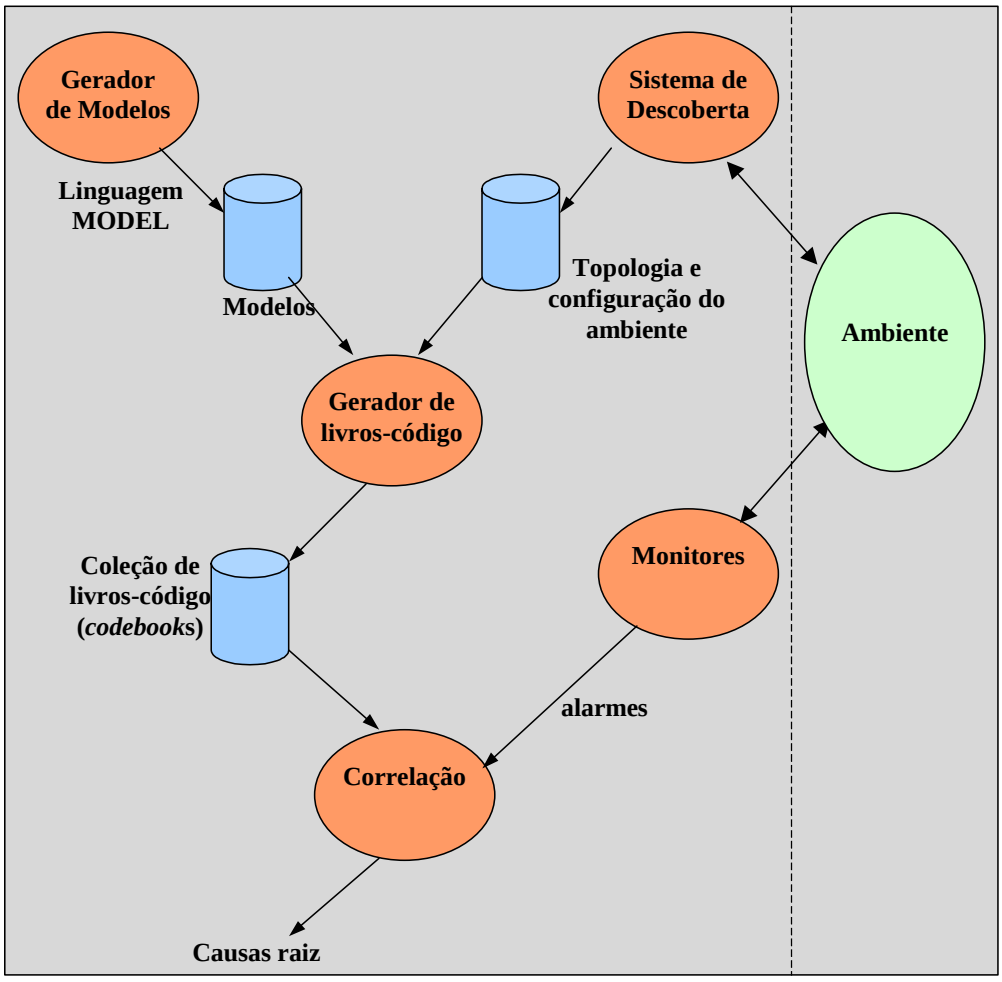


Figura 15 – Arquitetura geral do sistema SMARTS

Segundo (SMARTS, 2000), a análise de eventos é separada em três diferentes escopos:

> **Correlação intra-objeto**: Representa a correlação realizada no escopo de um elemento único, por exemplo, um equipamento. Ela permite a detecção da causa raiz relacionada a anomalias locais.
>
> **Correlação cruzada entre objetos**: Representa a correlação realizada sobre o escopo de diferentes elementos, porém relacionados dentro de um determinado domínio. Por exemplo, os efeitos de uma falha na interface de um roteador poderá gerar a indicação de perda de comunicação com este roteador mas também de outros roteadores e equipamentos. Assim, a necessidade da correlação cruzada entre objetos surge pelas dependências existentes entre entidades de um domínio.
>
> **Correlação cruzada entre domínios**: O último escopo de correlacionamento representa a correlação realizada através de domínios de gerenciamento. Considere-se o exemplo de correlação cruzada entre domínios na qual é observada uma falha de comunicação com um servidor. Isto pode ter sido causado pela falha de um roteador. A necessidade da correlação cruzada entre domínios decorrem da dependência existente entre os domínios de rede, servidores e de aplicação. Por exemplo: aplicações dependem de servidores que as executam, servidores dependem da infra-estrutura de comunicação que os conectam. Assim, é necessário correlacionar os resultados entre tais domínios.

O sistema SMARTS possui duas funcionalidade importantes associadas ao diagnóstico:

- **diagnóstico de anomalia**: a identificação da causa raiz;
- **análise de impacto de anomalia**: cada causa raiz geralmente causa vários efeitos no ambiente. A identificação do impacto é importante, pois permite determinar quais usuários e processos corporativos são afetados pela

anomalia. Isto possibilita notificar os usuários e disparar planos de contingência.

2.6.1 Modelo

Segundo Oshie (1997b), o modelamento de evento é um componente essencial em um sistema de correlação de eventos e é formado por dois componentes básicos:

- um modelo de evento;
- um algoritmo de raciocínio.

Em SMARTS, o modelo de evento é composto por um modelo de classe de eventos e uma topologia de objetos. O modelo de classe de eventos descreve as regras gerais de propagação de eventos de uma classe de objeto em outra, enquanto que a topologia de objetos descreve suas instâncias relacionadas ao ambiente, refletindo assim o estado corrente.

A linguagem MODEL (OSHIE 1997a; OSHIE 1997b) permite realizar a descrição dos objetos e dos eventos. É uma linguagem orientada a objetos completa, com suporte à herança e sobrecarga de métodos. Também fornece suporte para descrever como as observações devem ser realizadas utilizando gerenciamento SNMP.

Também fornece duas características essenciais à correlação de eventos. A primeira é a possibilidade de especificação declarativa de eventos utilizando expressões booleanas sobre valores do modelo de objeto. Isto permite que a definição do evento possa ser integrada ao modelo de objeto no qual o evento ocorre. A segunda permite ao usuário especificar regras de propagação de eventos, possibilitando a construção do grafo causal através da combinação do esquema de propagação de eventos e da topologia de objetos. Geralmente, os padrões de propagação de eventos dependem fortemente de como está configurada a topologia do ambiente.

O modelo de propagação de eventos é um modelo interpretativo comportamental e se mostra extremamente valioso para a geração de um modelo causal que, por sua vez, é utilizado para a geração dos livros-código

```
interface Iprouter: IP
    {
    instrumented attribute long ipInDiscards;
    instrumented attribute long ipOutDiscards;

    event PacketDiscardsHigh
        "O nível de discartes de pacotes está alto" =
       (delta ipInDiscards + delta ipOutDiscards) / delta_time
           > discardsTheshold;

    instrument SNMP;

    problem Congestion "High congestion" =
           PacketDiscardHigh 1.0, ConnectonPacketLoss 0.8;

    propagate symptom ConnectionPacketLossHigh =
        TransportConn, Undelying, PacketLossHigh;

    relationshipset Undelying, TransportConn, LayeredOver;
    }

interface TransportConn
    {
    propagate symptom PacketLossHigh =
        Port, ConnectedTo, PacketLossHigh;
    }

interface UDPPort: Port
    {
    propagate symptom PacketLossHigh =
        Appl, Undelying, PacketLossHigh;
    }
```

Figura 16 – Exemplo de descrição utilizando a linguagem MODEL.

A Figura 16 mostra um exemplo de descrição utilizando a linguagem MODEL. Pode-se perceber que MODEL fornece uma plataforma flexível para expressar a

propagação de eventos. Importante também é a possibilidade de reusar as definições de objetos.

2.6.2 Método de correlação por livro-código (*codebook*)

A correlação de eventos baseada na técnica de livro-código (*codebook*), também chamada de codificação, foi patenteada em 1994 e apresentada em (KLIGER, 1995) em 1995. Nesta técnica, os sintomas observados no ambiente são representados por um código que é comparado a um livro-código previamente compilado para a determinação da anomalia causadora dos sintomas observados. Dependendo das características do livro código é também possível ser resistente a ruído, ou seja, operar mesmo na presença de falta de sintomas ou sintomas espúrios. A principal vantagem apresentada é a velocidade de correlação. Porém, é tipicamente um sistema atemporal, não suportando composição com técnicas de correlação temporal.

O processo de geração do livro-código é apoiado no grafo de causalidade[13]. gerado a partir do modelo de eventos e topologia de objetos. Contudo, este grafo não se mostra apropriado para utilização. É necessário transformá-lo em um grafo de correlação bipartido, retirando sintomas indiretos e agregando ciclos. Um exemplo de grafo de correlação está mostrado no exemplo da Figura 17.

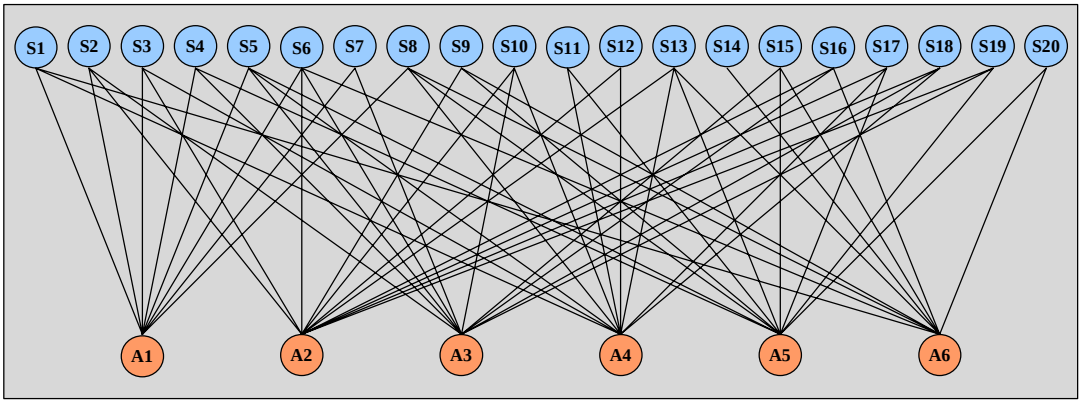


Figura 17 – Exemplo de grafo de correlação, derivado do exemplo de (KLIGER, 1995)

[13] No Anexo 2 é apresentada uma breve introdução a respeito do grafo de causalidade.

É utilizado o modelo determinístico de causalidade[14]. Desta forma, o grafo pode ser representado na forma de tabela chamada de matriz de correlação, como mostrado na Figura 18. O valor 1 indica a possibilidade de causalidade e 0 indica a impossibilidade de causalidade.

	Anomalia						ok
Sintoma	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	
S1	1	0	0	1	0	1	0
S2	1	1	1	1	0	0	0
S3	1	1	0	1	0	0	0
S4	1	0	1	0	1	0	0
S5	1	0	1	1	1	0	0
S6	1	1	1	0	0	1	0
S7	1	0	1	0	0	1	0
S8	1	0	0	1	1	1	0
S9	0	1	0	0	1	1	0
S10	0	1	1	1	0	0	0
S11	0	0	0	1	1	0	0
S12	0	1	0	1	0	0	0
S13	0	1	0	1	1	1	0
S14	0	0	0	0	0	1	0
S15	0	0	1	0	1	1	0
S16	0	1	1	0	0	1	0
S17	0	1	0	1	1	0	0
S18	0	1	1	1	0	0	0
S19	0	1	1	0	1	0	0
S20	0	0	0	0	1	1	0

Figura 18 – Exemplo de matriz de correlação derivada do grafo de correlação

A matriz de correlação explicita os vetores código que indicam a presença de uma única anomalia. Um vetor código representa a configuração das observações em um determinado momento indicando ou não a presença de sintomas. A Figura 19 mostra os vetores código para a matriz de correlação apresentada na Figura 18.

[14] No Anexo 2 é apresentada uma breve descrição do modelo determinístico de causalidade.

```
A1 = (1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0)
A2 = (0,1,1,0,0,1,0,0,1,1,0,1,1,0,0,1,1,1,1,0)
A3 = (0,1,0,1,1,1,1,0,0,1,0,0,0,0,1,1,0,1,1,0)
A4 = (1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,0,1,1,0,0)
A5 = (0,0,0,1,1,0,0,1,1,0,1,0,1,0,1,0,1,0,1,1)
A6 = (1,0,0,0,0,1,1,1,1,0,0,0,1,1,1,1,0,0,0,1)
```

Figura 19 – Vetores código da matriz de correlação da Figura 18

O problema da correlação utilizando a técnica de livro-código é determinar, dado um vetor código que representa os sintomas observados em um determinado momento, qual vetor da matriz de correlação é mais próximo. O vetor mais próximo indica qual anomalia é a mais provável. Por exemplo, seja o vetor código mostrado na Figura 20 que representa os sintomas observados no ambiente em um determinado momento.

```
obs= (0,0,0,1,1,0,0,1,1,0,0,1,0,1,0,1,0,1,1)
```

Figura 20 – Exemplo de vetor código derivado de uma observação

Este vetor é muito semelhante ao vetor código associado à anomalia A5, exceto por um sintoma. É possível a ocorrência de algum erro durante o processo de observação causando a ausência do sintoma S11. Este é um exemplo de ruído nas observações. Outra possibilidade é a ocorrência de uma falha no modelamento.

Geralmente, é possível a existência de sintomas que são redundantes. É o caso dos sintomas S2 e S3. Eles auxiliam o processo de correlação somente em relação à anomalia A3. Todavia, a anomalia A3 pode também ser discriminada por outros sintomas. É comum a existência de redundância nas matrizes de correlação. Um subconjunto muito menor de observações pode ser selecionado e mesmo assim possibilitar a distinção entre as anomalias. Este subconjunto de sintomas é chamado de "**livro-código**". A Figura 21 mostra um livro-código derivado da mesma matriz de correlação, contendo somente três sintomas, e que permite ainda a distinção entre as anomalias. Entretanto, se ocorrer um ruído e variar uma das observações, a correlação irá indicar uma resposta errada.

	Anomalia						ok
Sintoma	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	
S1	1	0	0	1	0	1	0
S2	1	1	1	1	0	0	0
S4	1	0	1	0	1	0	0

Figura 21 – Exemplo de livro-código de raio 0,5 (distância mínima=1), extraída de (KLIGER, 1995)

A distância entre os códigos das anomalias é medida pelo raio. O raio de um livro-código é, por definição, a metade da menor distância do código de Hamming[15] entre seus códigos. A Figura 22 mostra a distância de código de Hamming para a matriz de correlação original (apresentada na Figura 18). Nela, a menor distância de Hamming é 8, implicando um raio de livro-código de valor 4.

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**ok**
A1		13	8	9	12	11	8
A2			9	8	13	12	11
A3				13	12	13	10
A4					11	14	11
A5						9	10
A6							9

Figura 22 – Distância entre códigos para o exemplo da Figura 18

A Figura 23 mostra a distância entre códigos para o exemplo da Figura 21. A menor distância é 1, implicando em um raio 0,5.

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**ok**
A1		2	1	1	2	2	3
A2			1	1	2	2	1
A3				2	1	3	2
A4					3	1	2
A5						2	1
A6							1

Figura 23 – Distância entre códigos para o exemplo da Figura 21

Os livros código com raio de valor maior ou igual a 1 possui a propriedade de ser tolerante à alteração do valor de uma observação. A Figura 24 mostra um livro-

[15] O Anexo 3 apresenta uma breve descrição a respeito do código de Hamming.

código com raio 1,5. As distâncias de código Hamming estão mostradas na Figura 25.

	Anomalia						ok
Sintoma	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	
S1	1	0	0	1	0	1	0
S3	1	1	0	1	0	0	0
S4	1	0	1	0	1	0	0
S6	1	1	1	0	0	1	0
S9	0	1	0	0	1	1	0
S18	0	1	1	1	0	0	0

Figura 24 – Exemplo de livro-código de raio 1,5 (distância mínima=3), extraída de Klinger (1995)

	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**ok**
A1		4	3	3	4	3	4
A2			3	3	4	3	4
A3				4	3	4	3
A4					5	4	3
A5						3	2
A6							3

Figura 25 – Distância entre códigos para o exemplo da Figura 24

2.6.3 Principais vantagens

Segundo White (1998), as principais vantagens apresentadas pelo sistema SMARTS são:

- Indica com grande precisão a anomalia raiz em segundos;
- Realiza a correlação cruzada de dados e eventos de redes, sistemas e aplicações;
- Automaticamente computa e atualiza suas regras de correlação;
- Executa ordem de magnitude mais rápida que sistemas baseados em regras;
- Resistente a ruído, ou seja, tolera perda de alguns eventos (sintomas) e também tolera a ocorrência de alguns eventos espúrios.

Outra vantagem que pode ser mencionada é a possibilidade de seletivamente reduzir o número de observações necessárias para o correlacionamento.

2.6.4 Principais desvantagens

Ainda segundo White (1998), as principais desvantagens apresentadas pelo sistema SMARTS são:

- Trata somente falhas únicas;
- Para realizar o correlacionamento, todos os sintomas (eventos) devem estar disponíveis, ou seja, não suporta informações imprecisas;
- Requer o completo conhecimento das anomalias antes que o livro-código seja computado;
- Não suporta correlacionamento temporal, é essencialmente atemporal;
- Livro-código deve ser recompilado sempre que o ambiente é alterado.

Outra desvantagem pode ser mencionada: existem limitações quanto a resistência à ruído principalmente em anomalias que produzem somente um único sintoma.

2.7 Conclusão

Este capítulo apresenta parte da teoria associada aos sistemas de correlação e diagnóstico. Ao final é apresentado o sistema de diagnóstico SMART, que é um sistema que possui diversas características interessantes, sendo porém essencialmente atemporal.

3. Sistemas de diagnóstico baseado em modelo com dimensão temporal

Em um trabalho a respeito de diagnóstico baseado em modelo com dimensão temporal, Console (1998-b) apresenta a existência de pelo menos três dimensões importantes relacionadas em um sistema de diagnóstico baseado em modelo:

- Tarefa de diagnóstico: Como é a tarefa de diagnóstico e sua dinamicidade;
- Natureza do dispositivo. Cada sistema/dispositivo possui uma forma diferente de comportamento no tempo;
- Tipo de dado temporal. Como as observações (alarmes, estados, etc.) e os dados temporais estão disponíveis durante o processo de diagnóstico.

Estas dimensões influenciam tanto o modelamento quanto o raciocínio (processo de diagnóstico).

Particularmente, esta tese busca realizar o modelamento da observação de forma a adicionar informações a respeito de imprecisões temporais e inexistência de observação (intervalos de tempo de incerteza) que a possibilitem uma melhor precisão ao sistema de diagnóstico.

3.1 Modelagem temporal em sistemas baseados em modelo

Em outro trabalho relacionado a modelamento temporal em sistemas de diagnóstico baseados em modelo, Console (1998-a) apresenta uma classificação a respeito de possíveis formas de modelamento atemporal e temporal. São relacionadas as seguintes:

- diagnóstico atemporal sobre único instante;
- diagnóstico atemporal sobre coleção de sintomas;

- diagnóstico atemporal sobre múltiplos instantes;
- diagnóstico temporal;
- diagnóstico temporal variante no tempo.

3.1.1 Diagnóstico atemporal sobre um único instante

No diagnóstico atemporal sobre um único instante de observação (*atemporal single-snapshot*), os sintomas são observados em um único instante do tempo. Não existe qualquer informação temporal. A solução é o conjunto de anomalias que explica o sintoma. A Figura 26 ilustra como as observações são utilizadas por um sistema de diagnóstico atemporal sobre único instante.

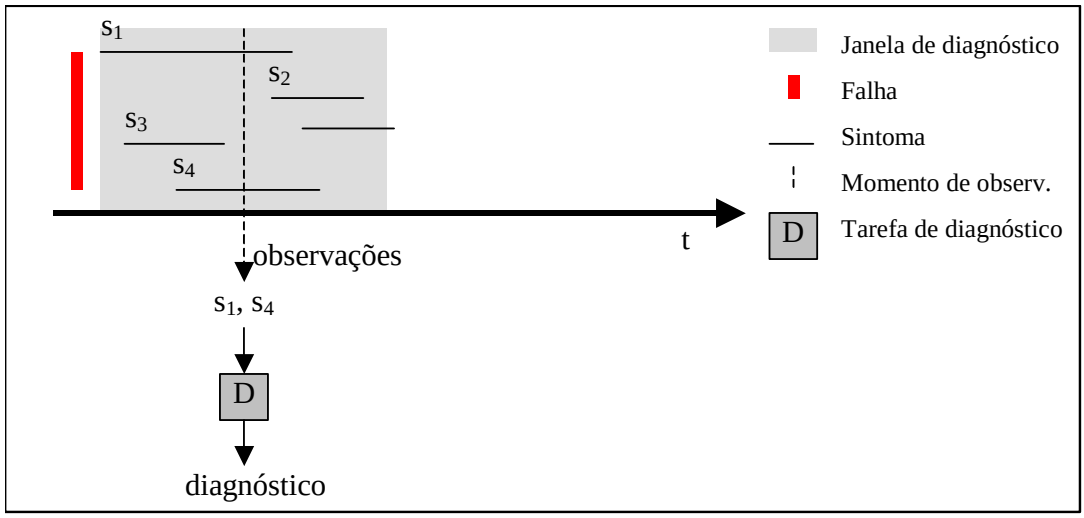


Figura 26 – Sistema de diagnóstico atemporal sobre único instante.

3.1.2 Diagnóstico atemporal sobre coleção de sintomas

No diagnóstico atemporal sobre coleção de sintomas (*atemporal sympton collection diagnosis*), o conjunto de sintomas que ocorre em uma janela temporal, gerado em diversos momentos de observação sobre esta janela, é utilizado como entrada para o sistema de diagnóstico. Estes sintomas não possuem nenhuma informação do momento de ocorrência, momento de observação ou duração. A solução é o conjunto de anomalias que explicam os sintomas observados. A Figura 27 ilustra como as observações são utilizadas por um sistema de diagnóstico atemporal sobre uma coleção de sintomas.

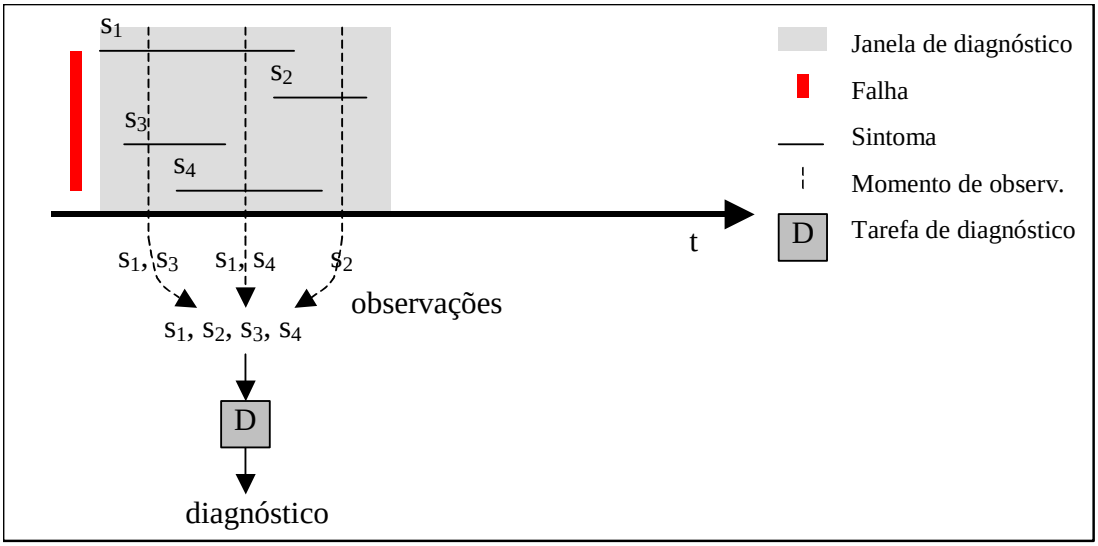


Figura 27 – Sistema de diagnóstico atemporal sobre uma coleção de sintomas.

3.1.3 Diagnóstico atemporal sobre múltiplos instantes

No diagnóstico atemporal sobre múltiplos instantes (*atemporal multiple-snapshot diagnosis*), as observações são realizadas em determinados instantes de janela de tempo, resultando em um conjunto de sintomas em cada momento de observação. O diagnóstico atua de forma independente, em cada instante de observação, utilizando os sintomas desse instante. O resultado do diagnóstico é a união do conjunto de anomalias que constitui a solução de cada instante. O diagnóstico é considerado atemporal pois a dimensão temporal é descartada: a ordem e a localização temporal dos sintomas não são utilizados efetivamente. A Figura 28 ilustra como as observações são utilizadas por um sistema de diagnóstico atemporal sobre múltiplos instantes.

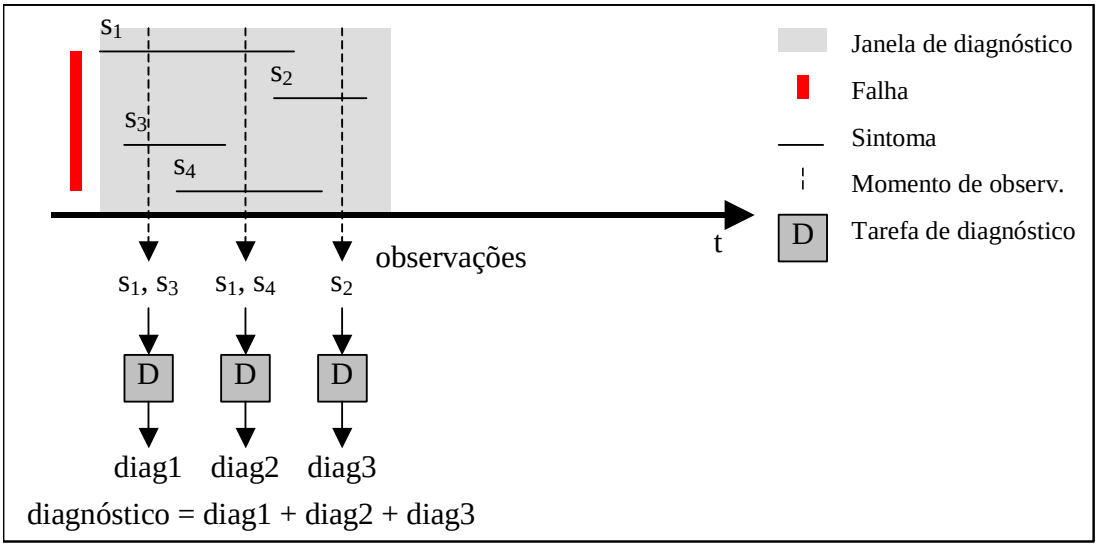


Figura 28 – Sistema de diagnóstico atemporal sobre múltiplos instantes.

3.1.4 Diagnóstico temporal

No diagnóstico temporal (*temporal diagnosis*) é modelado o comportamento dinâmico do dispositivo diagnosticado baseado no fato de que o sistema, estando em um determinado estado (normal ou anômalo), manifesta-se após algum tempo e com determinada duração.

O diagnóstico considera, não somente a observação, mas também sua localização temporal (ou seja o momento em que ocorreu).

Tipicamente, modela o relacionamento temporal da anomalia com os sintomas apresentados.

Em alguns casos este tipo de diagnóstico pode ser classificado como dinâmico, ou seja, leva em consideração o estado interno do sistema, que também precisa ser modelado. Nestes casos, a saída do sistema depende não somente da entrada mas também de um estado interno. O diagnóstico é realizado levando-se em consideração o estado interno no período de análise. A Figura 29 ilustra como as observações são utilizadas por um sistema de diagnóstico temporal.

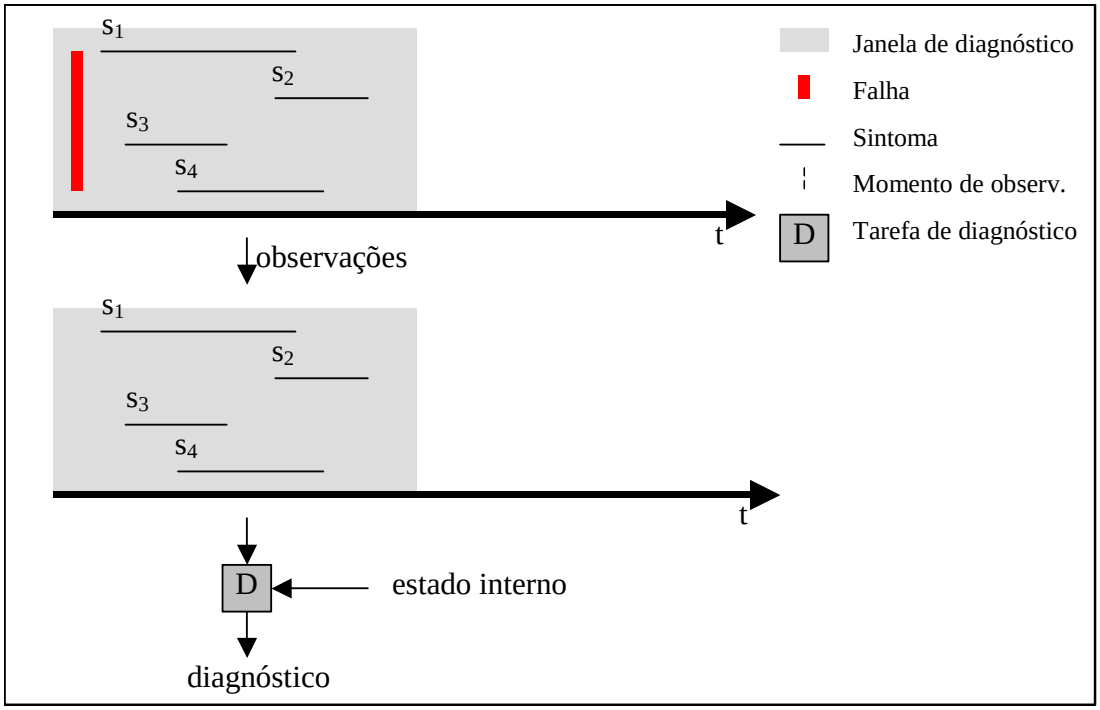


Figura 29 – Sistema de diagnóstico temporal.

3.1.5 Diagnóstico temporal variante no tempo

No diagnóstico temporal variante no tempo (*time-varying diagnosis*) é assumido que seja possível especificar quais transições entre anomalias são possíveis de ocorrer no sistema diagnosticado (nem todos os sistemas permitem tal modelamento), informando também quais são as restrições temporais entre elas. Por exemplo, "ocorrendo a anomalia 1, em seguida é possível ocorrer a anomalia 2 ou anomalia 3, …". A solução do problema do diagnóstico corresponde à reconstrução do histórico de anomalias do dispositivo para encontrar o conjunto de anomalias ou a anomalia raiz quando for o caso. Ou seja, a solução do problema é o conjunto de anomalias atribuídas a cada momento de forma que:

- a anomalia atribuída a cada momento explique os sintomas apresentados no momento;
- histórico de anomalias atribuídas é consistente com o modelo de comportamento do dispositivo na evolução do tempo.

A Figura 30 ilustra como as observações são utilizadas por um sistema de diagnóstico temporal variante no tempo.

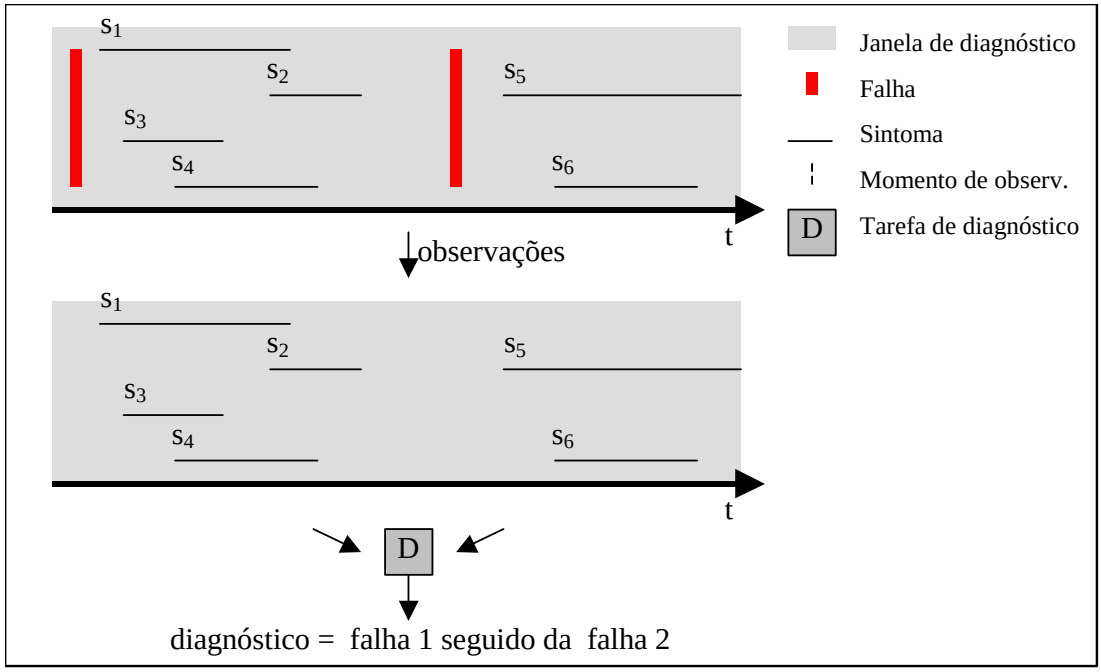


Figura 30 – Sistema de diagnóstico temporal variante no tempo.

Este é o tipo de modelamento temporal ideal para ser utilizado no diagnóstico de sistemas distribuídos porque permite utilização de informações sobre o relacionamento entre anomalias e anomalia e sintoma.

3.2 Ontologia do tempo

Ainda segundo Console (1998b), existem diferentes ontologias de tempo que podem ser adotadas no modelo e no processo de diagnóstico:

- tempo quantitativo;
- tempo qualitativo;
- tempo como uma seqüência de estados;
- abstrações Adhoc.

3.2.1 Tempo quantitativo

O modelo fornece restrições temporais quantitativas a respeito da evolução do sistema/dispositivo. Por exemplo, pode ser especificada a latência esperada entre a anomalia e seus efeitos ou a duração esperada de um sintoma.

3.2.2 Tempo qualitativo

Este é um modelo mais fraco que especifica somente restrições qualitativas (como as apresentadas em (ALLEN, 1983) a respeito da evolução do estado do comportamento de um dispositivo. Assim, pode ser especificado o relacionamento temporal entre anomalias e sintomas ou mesmo entre anomalias.

3.2.3 Tempo como uma seqüência de estados

É um modelo ainda mais fraco no qual o tempo é considerado como uma seqüência de estados (pontos) e o único tipo de relacionamento possível é em relação à ordem dos estados.

3.2.4 Abstrações *Ad hoc*

Ontologias especiais podem ser definidas levando em consideração aspectos dos três casos anteriores.

4. Interação do sistema de diagnóstico com o ambiente

Este capítulo discute algumas características importantes relacionadas aos sistemas de diagnóstico de ambiente distribuído. Particularmente:

- em relação à sua interação com o ambiente;
- em relação ao tempo.

A forma com que um sistema de diagnóstico opera e interage com o ambiente (ou com os sistemas de apoio, como os coletores de observações), causa impacto nos métodos de diagnóstico utilizados. Assim, algumas classificações que podem ser utilizadas **em relação à sua interação com o ambiente** são:

- quanto ao tipo da observação obtida;
- quanto ao controle do processo de observação.

Outras características que causam impacto nos métodos de diagnóstico estão associadas a **aspectos relacionados ao tempo**, particularmente:

- quanto ao instante ou janela de diagnóstico;

Este capítulo tem como objetivo apresentar algumas caracterizações de sistema de diagnóstico que serão utilizadas no decorrer do trabalho possibilitando também um melhor entendimento a respeito dos métodos de diagnóstico apropriados para cada caso.

4.1 Classificação quanto ao tipo da observação recebida

A qualidade da informação recebida pelo sistema de diagnóstico (observação) também é um fator importante que influencia a tarefa de diagnóstico.

Figura 31 – Observações em um sistema de diagnóstico

Em um ambiente computacional existem diversas classes de objetos gerenciados cujos valores podem ser classificados em[16]:

- valor bruto (*raw value*);
- estado.

Valor bruto: O valor de um objeto gerenciado é denominado bruto (*raw*) se individualmente não indica se o componente está em um estado normal ou anômalo. São tipicamente valores quantitativos. Na monitoração de um ambiente computacional distribuído é comum a existência de objetos gerenciados cujo valor é do tipo bruto. São os objetos quantitativos. Vários objetos gerenciados SNMP da MIB-II possuem valores brutos. A Figura 32 mostra exemplos de alguns destes.

```
10.0.0.1, mib-2.if.ifTable.ifEntry.ifInOctets.2 = 39928
10.0.0.1, mib-2.if.ifTable.ifEntry.ifInErrors.2 = 832
10.0.0.1, mib-2.if.ifTable.ifEntry.ifInDiscards.2 = 1983
10.0.0.5, mib-2.rmon,etherHistoryTable.etherHistoryEntry.
                              etherHistoryUtilization.5 = 35
```

Figura 32 – Exemplo de valores de objetos gerenciados SNMP da MIB-2 que não possuem significado isoladamente

Estado: Existem objetos gerenciados cujo valor representa diretamente um estado. São tipicamente valores qualitativos. Por exemplo, o objeto gerenciado SNMP `ifOperStatus` pode assumir os seguintes valores: “`up`”, “`down`” ou “`testing`”. Os valores “`down`” e “`testing`” representam estados anômalos para o componente, enquanto que “`up`” representa um estado normal.

[16] A seção 5.1.3 apresenta a definição de objeto gerenciado.

Um sistema de coleta de informações (por exemplo, uma plataforma de gerenciamento) é capaz de transformar um valor bruto em um valor tipo estado. É possível a definição de estados associados a um objeto gerenciado e a definição dos limiares de transição entre tais estados. Por exemplo, para um determinado objeto é possível definir dois estados: NORMAL e ANÔMALO, como mostrado na Figura 33.

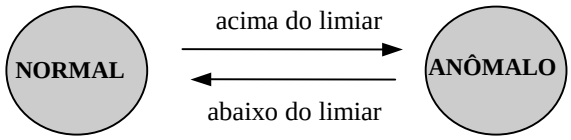


Figura 33 – Exemplo de mapeamento da observação em estados.

Exemplos de observação do tipo estado são:

- estado operacional = *up* (estado normal) ou *down* (estado anômalo);
- ocupação da memória = baixa (estado normal) ou alta (estado anômalo);
- octetos recebidos por uma interface = pouco (estado anômalo), médio (estado normal), alto (estado anômalo).

Estando definido o significado de um valor bruto e um valor tipo estado é possível classificar o tipo de observação recebida por um sistema de diagnóstico. Ele pode ser:

- observação tipo valor bruto (*raw value*);
- observação tipo estado;
- observação tipo transição de estado;
- observação tipo intervalo de tempo de ocorrência de estado.

4.1.1 Observação tipo valor bruto (*raw value*)

O tipo de observação obtido por um sistema de diagnóstico é classificado como sendo de “valor bruto” (*raw value*) quando o sistema de coleta de observações (geralmente a plataforma de gerenciamento) passa ao sistema de diagnóstico valores brutos.

Determinadas tarefas de diagnóstico, como é o caso do diagnóstico de falhas, geralmente se baseiam nos estados dos componentes (em geral estados normal e anômalo) para executar o raciocínio de diagnóstico. Um sistema de diagnóstico de falhas, portanto, necessita de um subsistema intermediário que transforme uma observação bruta em estados.

Já sistemas de diagnóstico do tipo predição geralmente necessitam de observações quantitativas (valor bruto ou semiprocessado) a fim de possibilitar, através de um histórico evolutivo, a previsão do comportamento do sistema. Por exemplo, prever a evolução da utilização de um determinado enlace de comunicação a fim de programar seu aumento de capacidade.

4.1.2 Observação tipo estado

O tipo de observação obtido por um sistema de diagnóstico é classificado como sendo “estado” quando o sistema de coleta de observações (geralmente uma plataforma de gerenciamento) passa ao sistema de diagnóstico o estado do objeto gerenciado.

4.1.3 Observação tipo transição de estado

O tipo de observação obtido por um sistema de diagnóstico é classificado como sendo “transição de estado” quando o sistema de coleta de observações (geralmente a plataforma de gerenciamento) informa ao sistema de diagnóstico somente as transições de estados.

Esta é uma das formas possíveis quando se utiliza uma plataforma de gerenciamento. Usualmente é possível definir diagramas de transição de estado a partir de observações de objetos. A cada transição pode ser gerado um alarme. Cabe à plataforma de gerenciamento observar o estado dos objetos gerenciados e gerar uma observação quando ocorrer uma transição de estado. Por exemplo, a plataforma poderá gerar um alarme quando passar do estado normal para anômalo e quando passar do estado anômalo para normal.

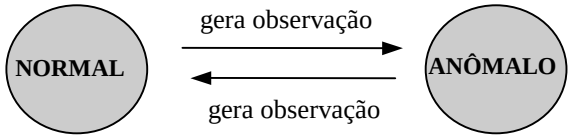


Figura 34 - Exemplo de mapeamento da observação em transição de estados

A Figura 35 mostra uma seqüência de transições de estados cuja observação foi obtida pelo recebimento de alarmes associados ao objeto gerenciado “alcançabilidade entre gerente e agente”.

severity	***alarm type***	***additional text***	***probable cause***
critical	*communications*	*unreacheble*	*loss of signal*
major	*communications*	*ip-primary-up*	
cleared	*communications*	*snmp-primary-up*	

Figura 35 – Exemplo de uma classe de alarmes gerados por uma plataforma de gerenciamento.

Este objeto possui três estados “unreacheble”, ip-primary-up” e “snmp-primary-up” e seu diagrama de transição de estados está mostrado na Figura 36.

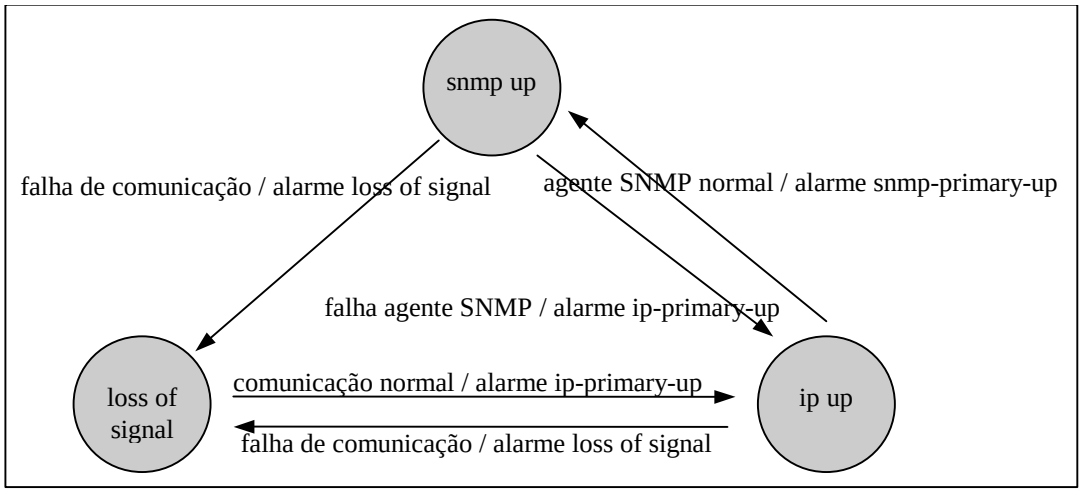


Figura 36 – Exemplo de diagrama de transição de estados.

4.1.4 Observação tipo intervalo de tempo de ocorrência de estado

A observação obtida por um sistema de diagnóstico é classificada como “intervalo de tempo de ocorrência de estado” quando a plataforma de gerenciamento (ou outro sistema coletor) informa ao sistema de diagnóstico os intervalos de tempo de

ocorrência de um determinado estado. O capítulo 6 descreve uma proposta de modelamento de observação com estas características.

4.2 Classificação quanto ao controle de processo de observações

4.2.1 Passivo puro

Um sistema de diagnóstico com controle de processo de monitoração passivo puro, como o próprio nome indica, aguarda de forma passiva a chegada das observações.

Geralmente, no momento de iniciação do sistema o sistema de diagnóstico informa ao subsistema de coleta de observações para realizar observações sobre o estado de determinados objetos gerenciados do ambiente. Porém, a partir deste momento, pouco controle possui a respeito da dinâmica deste processo.

As observações programadas devem ser aquelas associadas a todos os possíveis estados anômalos (sintomas) definidos, o que implica que o sistema coletor deve periodicamente consultar tais objetos gerenciados no ambiente. Para um ambiente com muitos equipamentos isto pode causar problemas de desempenho do sistema coletor ou problemas de contenção em enlaces de baixa capacidade.

O instante no qual uma observação é repassada ao sistema de diagnóstico depende da fase e período de amostragem utilizada pelo sistema coletor.

Quando o sistema coletor não é capaz de realizar uma observação, por exemplo quando ocorre uma perda de comunicação com o equipamento, o sistema coletor pode ou não informar este fato ao sistema de diagnóstico.

4.2.2 Ativo puro

Um sistema de diagnóstico com controle do processo de monitoração ativo puro, interage com o sistema coletor de observações para a realização de cada observação.

Um sistema de diagnóstico que opere no modo ativo puro é mais complexo devido ao fato de necessitar ter o controle das observações e instantes de requisição de cada

observação necessária. Por outro lado, possibilita ao sistema de diagnóstico a elaboração de tarefas de diagnostico mais sofisticadas, com controle dos instantes de observação.

Este modo de operação permite reduzir a quantidade de objetos observados do ambiente, diminuindo portanto o impacto da tarefa de diagnóstico no comportamento do sistema, além de diminuir os problemas associados à amostragem periódica (“*polling problem*”).

Este tipo de sistema possui um elevado grau de autonomia, sendo capaz de decidir e solicitar informações adicionais sobre o ambiente durante o processo de diagnóstico..

4.2.3 Semi-ativo

Um sistema de diagnóstico com controle do processo de monitoração semi-ativo mescla as características mencionadas anteriormente.

Por exemplo, pode operar inicialmente no modo passivo puro para a observação de objetos associados aos estados anômalos relacionados a sintomas iniciais e no modo ativo quando em uma tarefa de diagnóstico. Desta maneira, o sistema apenas detecta um sintoma inicial para ativar a sua tarefa de diagnóstico. A partir daí consulta os modelos e solicita, de forma ativa, informações a respeito de outros objetos.

4.3 Classificação quanto ao momento do diagnóstico

Em um sistema de diagnóstico, as tarefas de diagnóstico podem ou não ser ativadas imediatamente após o recebimento de um sintoma. Isto define algumas classes distintas de sistemas de diagnóstico:

- DMC – Diagnóstico de momento corrente;
- DMP – Diagnóstico de momento passado;
- DC – Diagnóstico contínuo.

O diagnóstico realizado no instante corrente, ou seja, nos casos em que é realizado imediatamente após a chegada de um sintoma, é sensivelmente prejudicado porque

grande parte do estado do ambiente está desatualizado. Isto decorre dos diversos problemas relacionados à técnica de amostragem[17]. Por este motivo, pode ser considerada a possibilidade de realização de diagnóstico após um determinado intervalo de tempo (janela temporal), garantindo que grande parte do estado do ambiente seja conhecida. Esta última forma de operação realiza o diagnóstico de um problema que ocorreu em um determinado momento no passado.

4.3.1 Diagnóstico de momento corrente (DMC)

O objetivo do "diagnóstico de momento corrente" (DMC) é descobrir as causas das anomalias do sistema que estão ocorrendo no momento atual (corrente). É importante lembrar que, devido ao problema de amostragem, as informações sobre o ambiente podem estar incompletas no momento exato da chegada do primeiro sintoma.

Este modo de operação é interessante para a emissão de um diagnóstico preliminar. Nesta situação, o sistema opera com um grau de erro certamente maior. Porém, emite imediatamente um diagnóstico sobre as possíveis causas de anomalias do sistema. Se o sistema de diagnóstico operar no modo ativo, possibilita a requisição de observações adicionais imediatamente.

4.3.2 Diagnóstico de momento passado (DMP)

O objetivo do "diagnóstico de momento passado" (DMP) é descobrir as causas das anomalias que ocorreram no sistema em um determinado momento no passado. Este diagnóstico é mais preciso que o anterior já que possui uma quantidade maior de informações do ambiente devido ao problema de amostragem (quanto mais no passado, mais completas as informações).

Porém, limita os resultados de um diagnóstico no modo ativo pois as anomalias podem não estar mais presentes, o que pode ser uma desvantagem. É importante, portanto, que todos os sintomas modelados sejam observados de antemão.

---

[17] Os problemas decorrentes do processo de amostragem são detalhados na seção 6.2.

Pode ser interessante ao sistema considerar a utilização dos dois modos de operação (DMC e DMP), um para emissão de um diagnóstico imediato e preliminar, e outro para um diagnóstico estável e com menor possibilidade de erros.

4.3.3 Diagnóstico contínuo (DC)

O diagnóstico contínuo (DC), geralmente utilizado pelos sistemas de diagnóstico temporal variante no tempo, não é focado em um momento específico. Utiliza o conhecimento sobre a localização temporal dos estados do ambiente para gerar as hipóteses para tais anomalias. A chegada de novas informações pode, inclusive, contradizer, e conseqüentemente destruir, uma hipótese (diagnóstico) gerada no passado.

5. Capítulo 5 - Anomalias, sintomas e suas relações

Um dos principais relacionamentos utilizados em um sistema de diagnóstico é a relação causal. Este capítulo discute a respeito do relacionamento causal entre anomalias e entre anomalias e sintomas.

Contudo, inicialmente é necessário uniformizar e formalizar algumas definições utilizadas no decorrer desta tese como: anomalia, objeto gerenciado, objeto intermediário, observação e sintoma, entre outros.

5.1 Do objeto gerenciado ao sintoma

A seguir são apresentadas diversas definições, algumas delas adaptadas para o contexto do ambiente diagnosticado, alvo deste trabalho: um ambiente computacional distribuído. Outras foram propostas principalmente devido à ausência de terminologia na literatura.

5.1.1 Componente

A função de um sistema de diagnóstico é identificar quais componentes do ambiente diagnosticado podem estar apresentando problemas em um determinado momento.

Definição 3: Componente

Componente é qualquer entidade existente no ambiente, seja físico (*hardware*), *software* (sistema operacional, processo ou módulo de *software*) ou abstrato (como domínio de repetição, domínio de *broadcast*, subrede, etc).

5.1.2 Anomalia

O problema apresentado por uma entidade é denominado de “anomalia”. A maior parte dos trabalhos na literatura utiliza o termo “falha”. Porém, neste trabalho falha será considerada um dos tipos de anomalia.

A palavra “falha” significa “*falta, defeito*”. Na literatura, o termo “falha” é utilizado de uma maneira mais genérica, e muitas vezes não apropriada para algumas áreas de gerenciamento como gerenciamento, de configuração e gerenciamento de segurança.

Definição 4: Anomalia

Uma anomalia é um estado do componente que indica um problema - um comportamento não esperado deste componente.

De acordo com esta definição, podem ser consideradas anomalias:

- falha: perda completa da funcionalidade;
- degradação de desempenho;
- erros de configuração;
- eventos de segurança.

Segundo Rose (1996) os eventos associados a uma anomalia (denominados “falhas” em seu trabalho) que ocorrem em um ambiente computacional podem ser classificados como:

- condição de problema: indica um problema que requer atenção;
- condição não usual: pode ocorrer em freqüência baixa. Se ocorrer em alta freqüência pode indicar um problema;
- condição associada à carga de utilização: sobrecarga ou sub-carga.

5.1.3 Objeto gerenciado

Todo componente do ambiente possui um comportamento (ou estado). Alguns destes estados podem ser observados por “entidades de *software*”. Em um sistema distribuído, estas entidades de software são geralmente os agentes de gerenciamento.

O papel de um agente, como mostrado na Figura 37, é realizar o mapeamento do comportamento do objeto real em valor de objeto gerenciado.

Definição 5: Classe de Objeto Gerenciado (COG).

É chamado de "Classe de Objeto Gerenciado", qualquer comportamento associado a uma classe de componente que seja passível de ser observado computacionalmente, ou seja, qualquer característica que possa ser representada por um valor de estado.

Esta definição deriva diretamente da definição utilizada no protocolo de gerenciamento SNMP (RFC1155; RFC1157; RFC1212; RFC1212; RFC1213; RFC1214; RFC1215) e foi utilizada para permitir uma uniformidade dos termos utilizados. No gerenciamento OSI (BRISA, 1993) a classe de objeto gerenciado aqui denotada é equivalente ao atributo de uma classe de objeto gerenciado.

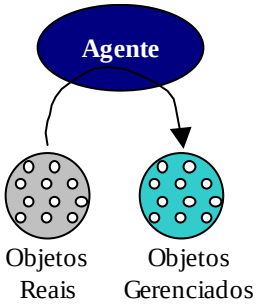


Figura 37 – O papel do agente de gerenciamento.

Associados a um componente podem existir um, dois, ou mesmo diversos objetos gerenciados. Pode também não apresentar nenhum objeto gerenciado. Este é o caso, por exemplo, de um cabo físico de rede, cujo comportamento pode ser inferido ou suposto através dos objetos gerenciados associados a outros componentes como, por exemplo, o estado do enlace ou a taxa de recepção associada à interface de rede ao qual o cabo esta conectado. Se existir taxa de recepção é possível inferir que o cabo não está rompido.

É possível citar como exemplos de COGs:

- número de octetos transmitidos por uma interface;
- estado do enlace de uma interface de comunicação;
- um atributo de uma classe de objeto gerenciado OSI;

- as classes de objetos gerenciados SNMP, por exemplo:
    - mib-2.system.sysName;
    - mib-2.if.ifTable.ifEntry.ifOperStatus;
    - mib-2.if.ifTable.ifEntry.ifInOctets;
    - mib-2.if.ifTable.ifEntry.ifInErrors;
    - mib-2.if.ifTable.ifEntry.ifInDiscards.

Definição 6: Objeto Gerenciado (OG)

Um "Objeto Gerenciado" é uma instância de uma "Classe de Objeto Gerenciado". Podem existir inúmeros objetos gerenciados derivados de uma mesma classe.

Exemplos de objetos gerenciados são:

- número de octetos transmitidos pela interface eth4 do roteador Jupter;
- estado do enlace da interface eth4 comunicação do servidor marte;
- um atributo de um objeto gerenciado OSI;
- os objetos gerenciados SNMP, por exemplo:
    - 10.0.0.1:mib-2.system.sysName;
    - 10.0.0.1:mib-2.if.ifTable.ifEntry.ifOperStatus.2;
    - 10.0.0.1:mib-2.if.ifTable.ifEntry.ifInOctets.2;
    - 10.0.0.1:mib-2.if.ifTable.ifEntry.ifInErrors.2;
    - 10.0.0.1:mib-2.if.ifTable.ifEntry.ifInDiscards.2.

Um OG pode ser obtido de diversas maneiras em um sistema de computação:

- envio de pacotes ICMP echo (*ping*) para equipamentos remotos;
- ativação remota de requisições para agentes de gerenciamento (SNMP, CMIP, proprietários, etc.);
- recebimento de alarmes de agentes de gerenciamento (SNMP, CMIP, proprietários, etc.);

- ativação remota de *scripts* ou utilitários através de acessos TELNET, rsh, rlogin, SSH entre outros;
- ativação local de utilitários que realizam ativações remotas com protocolos específicos como, por exemplo, rpcinfo;
- recebimento de "*trouble-tickets*";
- inserção manual de uma observação realizada pelo operador.

Definição 7: Valor de objeto gerenciado

Associado a um objeto gerenciado existe um valor que representa o estado aproximado do objeto real em um determinado instante.

O estado de um objeto gerenciado é representado por um valor. A seguir, são apresentados alguns exemplos:

- número de octetos transmitidos pela interface eth4 do roteador Jupter = 83772;
- estado do enlace da interface eth4 comunicação do servidor marte = "up";
- o valor de um atributo de um objeto gerenciado OSI;
- os objetos gerenciados SNMP, por exemplo:
    - 10.0.0.1, mib-2.system.sysName = "jupter";
    - 10.0.0.1.mib-2.if.ifTable.ifEntry.ifOperStatus.2 = "up";
    - 10.0.0.1, mib-2.if.ifTable.ifEntry.ifInOctets.2 = 39928;
    - 10.0.0.1, mib-2.if.ifTable.ifEntry.ifInErrors.2 = 832;
    - 10.0.0.1, mib-2.if.ifTable.ifEntry.ifInDiscards.2 = 1983;
- alcançabilidade do equipamento 192.168.30.14 = "reachable";
- latência ICMP echo entre o gerente e host 192.168.30.14 = 273 ms;

Definição 8: COG composta no tempo

Uma COG composta no tempo é também uma COG cujo valor é o resultado de uma expressão envolvendo a mesma ou outras COGs em momentos diferentes.

É muito comum a existência de COG cujo valor do OG associado tenha pouca valia isoladamente, tanto para um sistema de monitoramento quanto de diagnóstico, devido ao tipo de informação de estado disponibilizada. É o caso de COG cujo comportamento é reportado através de acumulador (contadores progressivos). No gerenciamento SNMP existem diversos objetos gerenciados que são definidos por acumuladores. Um exemplo típico é o COG SNMP mib-2.if.ifTable.ifEntry.ifInOctets que representa a quantidade acumulada de octetos recebidos por uma interface de comunicação desde que o agente de monitoramento (ou sistema operacional) iniciou sua atividade.

```
mib-2.if.ifTable.ifEntry.ifInOctets = 493834
```

Figura 38 – Exemplo de valor de objeto gerenciado sem significado isoladamente.

O exemplo da Figura 38 indica que desde que o equipamento foi ligado até o momento da consulta do estado do objeto foram recebidos 493834 octetos. Este valor não possui nenhum significado isoladamente. Nesta situação, uma informação de estado mais significativa poderia ser obtida através da composição de duas amostragens do OG em instantes distintos. Sabendo-se o intervalo de tempo entre as amostragens seria possível definir para este período a taxa média de transmissão em octetos/s, como mostrado no exemplo da Figura 39.

```
amostragem i  : mib-2.if.ifTable.ifEntry.ifInOctets= 354930000000
amostragem i+1: mib-2.if.ifTable.ifEntry.ifInOctets= 354939583487
intervalo de tempo entre amostragens: 100 s
taxa de recepção = 95835 octetos/s
```

Figura 39 – Exemplo de objeto gerenciado composto no tempo.

Portanto, podem existir COGs que são derivadas de expressões sobre outras COGs. Estas são chamadas de COGs compostas no tempo.

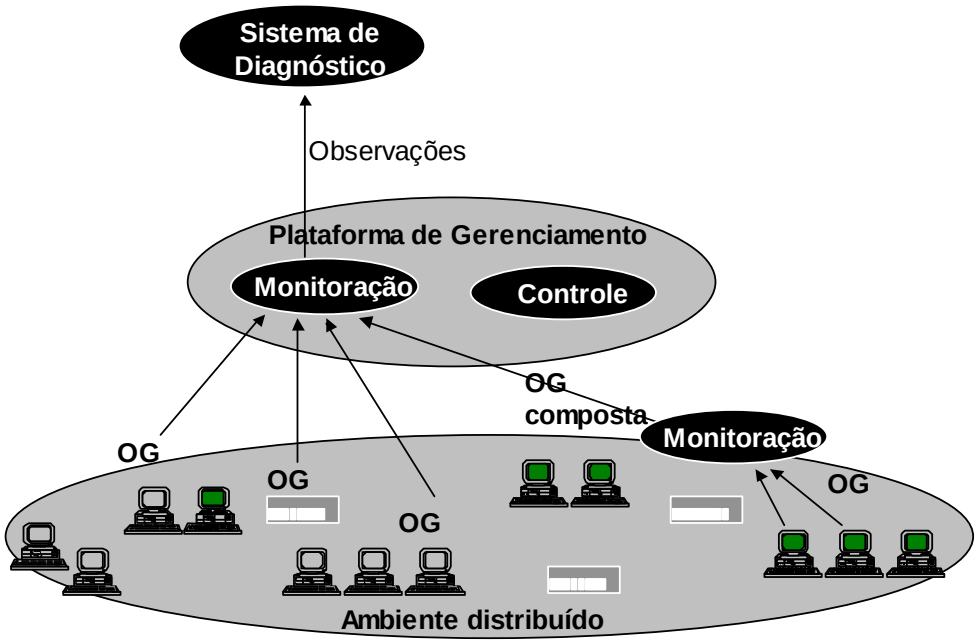


Figura 40 – Visão geral da arquitetura de um sistema de diagnóstico tradicional.

5.1.4 Objeto intermediário

O sistema de monitoração é o elo entre o ambiente e o sistema de diagnóstico. É responsável pela monitoração dos objetos gerenciados e, muitas vezes, também pelo tratamento destas informações a fim de torná-las adequadas a um sistema de diagnóstico. Em alguns ambientes este tratamento pode ser executado por um módulo à parte.

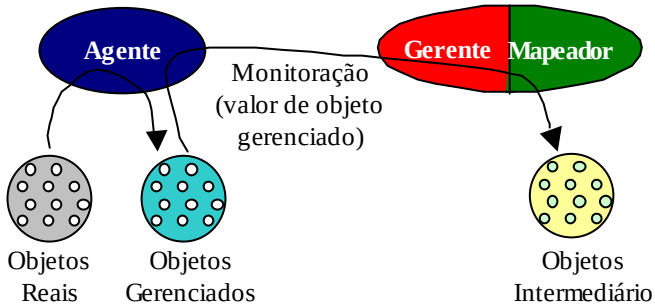


Figura 41 - Papel do gerente na obtenção de estados (valores) dos objetos intermediários.

A manipulação destes valores pelo sistema de monitoração pode levar ao surgimento de uma nova classe de objeto com propriedades e estados distintos do objeto

gerenciado associado original. Considere-se o exemplo de valor de objeto gerenciado, mostrado na Figura 42.

```
Taxa de transm. da interface eth0 do servidor jupter= 95,8 Kbps
```

Figura 42 – Exemplo de objeto gerenciado e seu valor.

Este valor, para um sistema de diagnóstico, pode ser ainda considerado um valor bruto e pode não ser útil. Em alguns casos poderia ser mais útil informar se a taxa está em uma situação normal ou anômala.

Neste caso, poderia existir uma representação intermediária do objeto indicando o seu estado, como mostrado na Figura 43.

```
Taxa de transm. da interface eth0 do servidor jupter = NORMAL
```

Figura 43 – Exemplo de representação intermediária e seu valor

Assim, surge a necessidade da definição deste novo objeto denominado “objeto intermediário”.

Definição 9: Classe de objeto intermediário (COI)

Uma “Classe de Objeto Intermediário” representa um aspecto do comportamento do sistema diagnosticado mantido pelo sistema de monitoração (geralmente para uma outra entidade externa, como um sistema de correlação ou de diagnóstico). Pode ser um mapeamento direto de uma classe de objeto gerenciado ou pode ser resultado de um processamento sobre uma ou mais classes de objetos gerenciados.

Uma instância de uma classe de objeto intermediário é denominada “Objeto intermediário”.

Definição 10: Objeto Intermediário (OI)

Um “Objeto Intermediário” é uma instância de uma “Classe de Objeto Intermediário”.

Assim como os objetos gerenciados, os intermediários também possuem valores associados.

Definição 11: Valor de objeto intermediário

Associado a um objeto intermediário existe um valor que representa o comportamento aproximado de um aspecto do ambiente em um determinado instante.

Mesmo quando o objeto intermediário é resultante de um mapeamento direto do objeto gerenciado, podem existir diferenças significativas. Um dos motivos da existência destas diferenças é decorrente do processo de amostragem. Se o valor do objeto intermediário for baseado em amostragem, ele, ao longo do tempo, será uma aproximação do valor (estado) do objeto gerenciado associado. A Figura 44 mostra um exemplo de um objeto intermediário baseado no objeto gerenciado SNMP mib-2.if.ifTable.ifEntry.ifOperStatus.

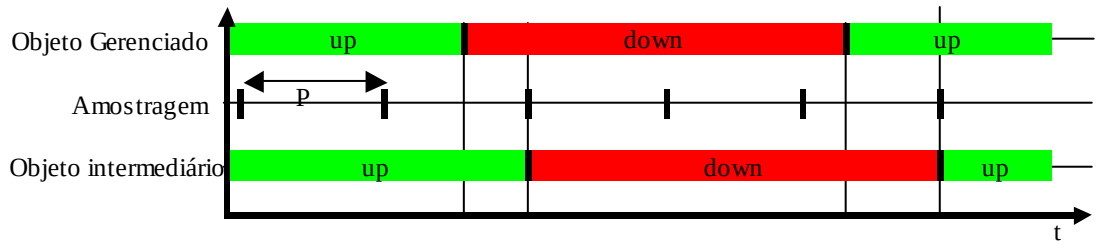


Figura 44 – Exemplo do objeto da diferença do estado observado no objeto intermediário em relação ao objeto gerenciado.

O sistema de monitoração pode, desta forma, disponibilizar para outros sistemas (como sistema de correlação ou sistema de diagnóstico) um conjunto específico de objetos, baseado nos estados dos objetos gerenciados, chamados de objetos intermediários.

Definição 12: COI composta no tempo

Uma COI composta no tempo é o resultado de uma expressão envolvendo um objeto gerenciado (ou mais de um objeto gerenciado) em momentos diferentes.

Assim como existem COG compostas no tempo, existem também as COI compostas no tempo.

5.1.5 Observação

O Sistema de Diagnóstico necessita ser alimentado com informações a respeito do estado do ambiente distribuído. Com base nessas informações será capaz de emitir um diagnóstico na eventualidade de ocorrência de anomalias.

Definição 13: Observação

Uma observação é um valor recebido pelo sistema de diagnóstico que representa um estado de um objeto intermediário que, por sua vez, representa uma aproximação de um determinado aspecto do comportamento do sistema.

Atenção: Neste trabalho, o termo "observação" será utilizado predominantemente para indicar as informações consumidas pelo sistema de diagnóstico.

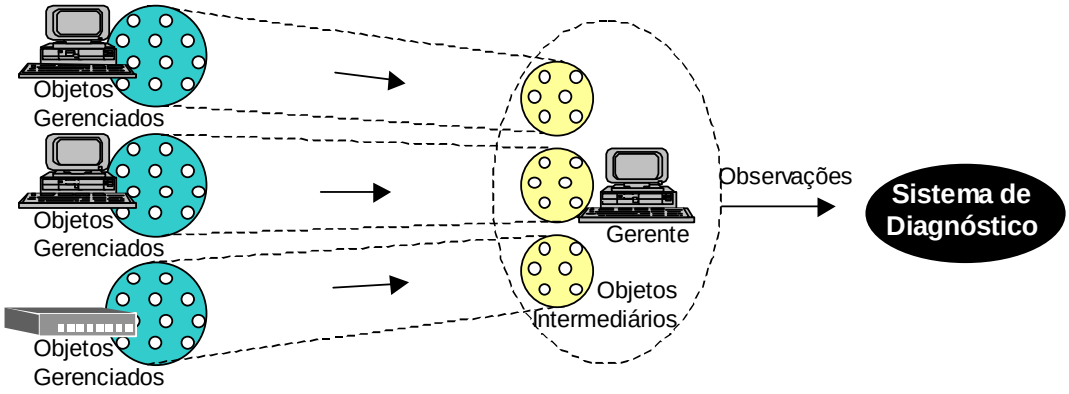


Figura 45 – Observações em um sistema de diagnóstico.

O Sistema de Diagnóstico pode ser alimentado com observações utilizando diferentes formas. No capítulo 4, é realizada uma classificação das diferentes formas de interação do sistema de diagnóstico com o sistema de coleta de informações e dos diferentes tipos de observação que podem existir.

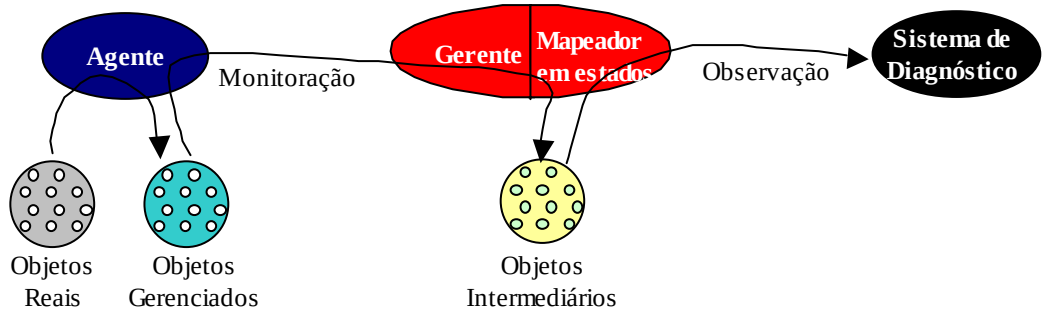


Figura 46 – Arquitetura tradicional de infra-estrutura para um sistema de diagnóstico

A Figura 46 mostra a arquitetura tradicionalmente utilizada como infra-estrutura para um sistema de diagnóstico.

Tradicionalmente, em sistemas de diagnóstico de ambientes distribuídos apoiados sobre redes de dados uma observação geralmente representa o estado de um objeto intermediário. Já em sistemas de diagnóstico de ambiente de telecomunicações uma observação geralmente representa uma transição de estado.

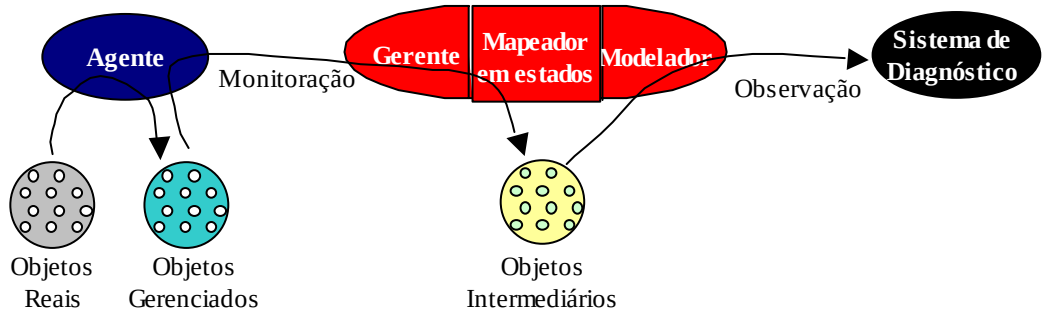


Figura 47 – Arquitetura da infra-estrutura para diagnóstico com o modelador de observações.

A presente tese propõe uma infra-estrutura para sistema de diagnóstico ligeiramente diferente da tradicional, a fim de possibilitar agregar informações sobre imprecisão temporal. Esta arquitetura está mostrada na Figura 47. Nela, a observação representa o intervalo aproximado de tempo no qual um determinado estado ocorre no objeto real. O modelamento da observação é detalhado no capítulo 6.

5.1.6 Sintoma

Existe um determinado tipo de observação, denominada sintoma, que é especialmente importante em um sistema de diagnóstico.

Definição 14: Sintoma

Um sintoma é qualquer observação que representa um estado anômalo do objeto intermediário.

O termo "sintoma" é descrito pelo dicionário (FERREIRA, 1986) como: "*1. Medicina, qualquer fenômeno ou mudança provocada no organismo por doença, e que, descritos pelo paciente, auxiliam a estabelecer um diagnóstico. 2. Sinal, indício. 3. Presságio, pressentimento, agouro*".

Benjamins (1993), em seu trabalho sobre métodos de solução de problemas para diagnóstico, descreve sintoma com sendo uma "*observação que desvia da expectativa*".

Um sintoma é importante pois pode indicar um comportamento anômalo em algum componente do ambiente. Mas a existência de um sintoma não indica necessariamente a ocorrência de uma anomalia. Por exemplo, um sintoma informando uma alta taxa de transmissão por uma interface de rede pode ser aceitável desde que não influa na comunicação.

5.1.7 Exemplo

A Figura 48 apresenta um exemplo mostrando um relacionamento das definições apresentadas nesta subseção.

Definição	**Exemplo**
Componente (Objeto Real)	Área de *swap*
Classe de Objeto Gerenciado	Tamanho total da área de *swap*
	Área de *swap* utilizada
Objeto Gerenciado	Tamanho total da área de *swap* do servidor júpiter
	Área de *swap* utilizada no servidor júpiter
	Tamanho total da área de *swap* do servidor marte
	Área de *swap* utilizada no servidor marte
Valor do Objeto Gerenciado	Tamanho total da área de *swap* do servidor júpiter = 100 Mbytes
	Área de *swap* utilizada no servidor júpiter = 90 Mbytes
	Tamanho total da área de *swap* do servidor marte = 100 Mbytes
	Área de *swap* utilizada no servidor marte = 40 Mbytes
Classe de Objeto Intermediário	Taxa de ocupação da área de *swap* = {NORMAL, QUASE_CHEIO, CHEIO }
Objeto Intermediário	Taxa de ocupação da área de *swap* do servidor júpiter

	Taxa de ocupação da área de *swap* do servidor marte
Valor do Objeto Intermediário	Taxa de ocupação da área de *swap* do servidor júpiter = QUASE_CHEIO
	Taxa de ocupação da área de *swap* do servidor marte = NORMAL
Observação	Taxa de ocupação da área de *swap* do servidor júpiter = QUASE_CHEIO
	Taxa de ocupação da área de *swap* do servidor marte = NORMAL
Sintoma	Área de *swap* do servidor júpiter está quase cheia.

Figura 48 – Exemplo de relacionamento das definições apresentadas.

5.2 Caracterização de uma relação causal

Uma relação causal[18] pode ser classificada de diversas maneiras. Neste trabalho, duas classificações são importantes:

- classificação quanto à possibilidade de causar um efeito;
- classificação quanto ao retardo do efeito.

5.2.1 Classificação quanto à possibilidade de causar um efeito

Seja uma relação causal C→ E, sendo "C" a causa e "E" um de seus efeitos. Esta relação pode ser classificada quanto a possibilidade de causar um efeito em:

- **Necessariamente causa**: quando da ocorrência da causa "C", sempre ocorre o efeito "E".
- **Possivelmente causa**: quando da ocorrência da causa "C", o efeito "E"pode ou não ocorrer.

A Figura 49 mostra exemplos de relações "necessariamente causa" e "possivelmente causa".

[18] O anexo 2 apresenta uma breve introdução às relações causais.

```
Computador_desligado necessariamente causa Processo_terminado
Partição_cheia       possivelmente causa   Processo_terminado
```

Figura 49 – Exemplos de relações “necessariamente causa” e “possivelmente causa”.

5.2.2 Classificação quanto ao retardo do efeito

A relação causal pode também ser classificada quanto ao retardo da ocorrência do efeito em relação à causa. Pode ser classificada quanto ao início do efeito em:

- início imediato;
- início retardado.

Também pode ser classificada quanto ao término do efeito em:

- término imediato;
- término retardado;
- término indeterminado.

Assim, existem seis combinações possíveis. A Figura 50 mostra, para cada combinação, a restrição temporal entre estes intervalos baseada na teoria de intervalos de tempo convexos[19] definida em Allen (1984).

[19] O Anexo 4 apresenta uma breve descrição a respeito de algumas formas de representação de tempo.

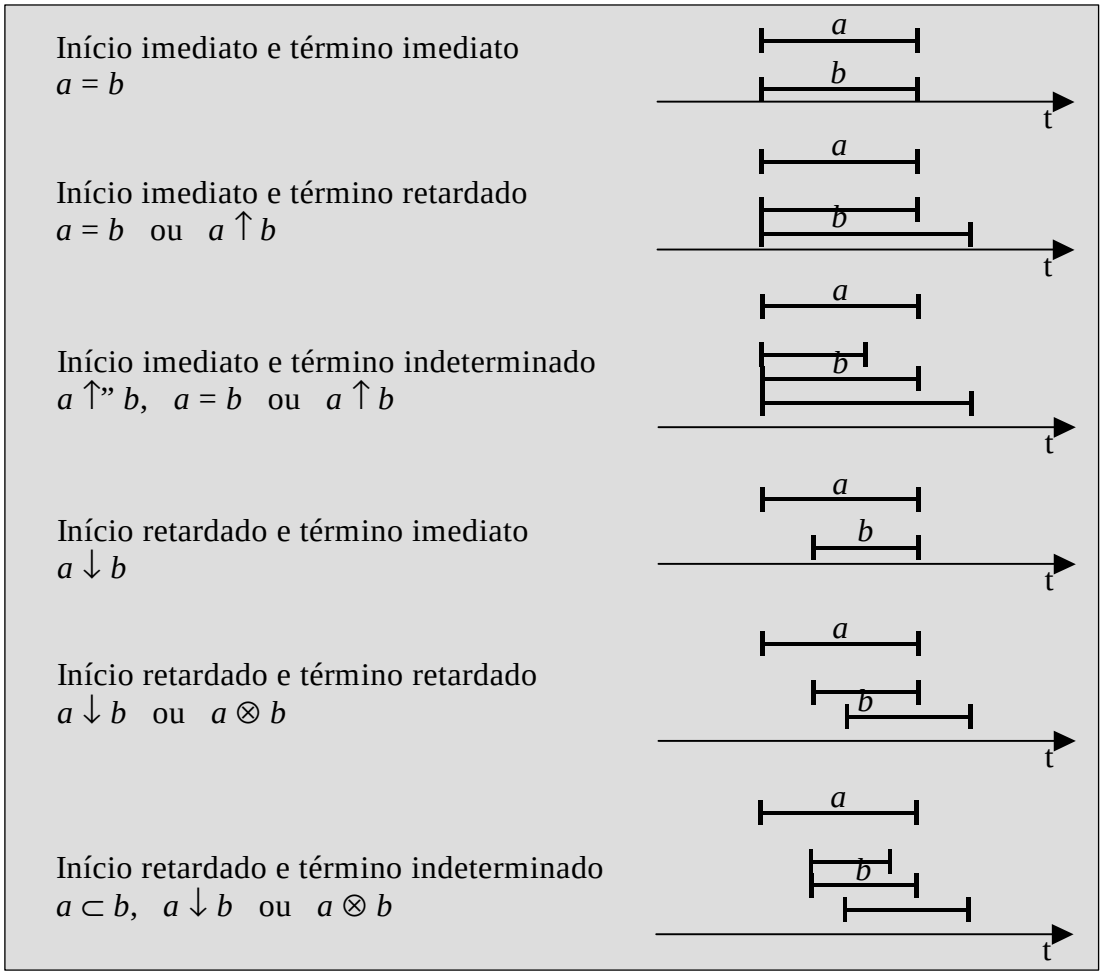


Figura 50 - As diferentes combinações dos intervalos de tempo entre causa e efeito.

Observação: Em filosofia é suposto implicitamente que o início de um efeito sempre se dê durante a ocorrência de sua causa. Porém, isto não é necessariamente verdadeiro quando é utilizado um modelo representativo. Por exemplo, sejam as relações causais de início e término retardado mostradas na Figura 51.

A → B
B → C

Figura 51 – Exemplo de relações causais de início e término retardado.

Em um modelo interpretativo, pode não ser adequado representar tal conhecimento dessa forma. Isto ocorre quando se deseja uma representação simplificada ou em um nível mais alto de abstração. Uma forma alternativa de representação seria a apresentada na Figura 52.

```
A → C
```

Figura 52 – Exemplo de representação alternativa para as relações da Figura 51.

A relação causal, quando representada desta forma, não possui necessariamente a propriedade “do início de um efeito sempre ocorrer durante a ocorrência sua causa”.

5.2.3 Relação causal entre anomalias e entre anomalia e sintoma

No modelo de sistema de diagnóstico proposto existem dois conjuntos nos quais as relações causais são aplicadas:

- A: o conjunto de anomalias;
- S: o conjunto de sintomas.

Assim, são definidos dois conjuntos de relações:

- $\Re_{AA} \subseteq A \times A$, o conjunto de relações causais entre anomalias;
- $\Re_{AS} \subseteq A \times S$, o conjunto de relações causais entre anomalias e sintomas.

5.3 Relacionamento causal direto entre anomalias

Em qualquer sistema, geralmente a ocorrência de uma determinada anomalia pode acarretar a ocorrência de outras anomalias. Este relacionamento é chamado de relacionamento causal direto entre anomalias e pode ser representado por $\Re_{AA} \subseteq A \times A$, sendo A o conjunto de anomalias. A Figura 53 ilustra uma relação causal A → B entre anomalias e sua relação inversa.

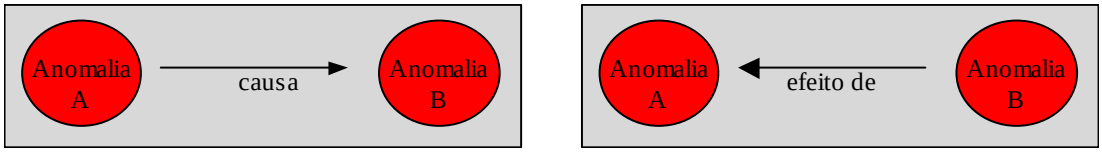


Figura 53 – Exemplo de relação causal entre anomalias

A anomalia “A” poderia ser “MEMÓRIA_CHEIA” e a anomalia B “PROCESSO_TERMINADO”. A Figura 54 mostra alguns exemplos deste relacionamento.

```
MEMÓRIA_CHEIA         causa PROCESSO_TERMINADO
VENTILADOR_CPU_FALHO  causa ALTA_TEMPERATURA_CPU
ALTA_TEMPERATURA_CPU  causa FALHA_NO_EQUIPAMENTO
FALHA_NO_EQUIPAMENTO  causa PROCESSO_TERMINADO
```

Figura 54 – Exemplo de relações causais entre anomalias

Em um ambiente de computação, a ocorrência de uma anomalia pode:

- não causar nenhuma outra anomalia;
- causar uma outra anomalia;
- causar mais que uma anomalia.

Se forem conhecidas as possíveis anomalias que um determinado sistema pode apresentar, bem como o relacionamento causal entre elas, é possível construir o conjunto de relações causais, como ilustrado na Figura 55 e Figura 56.

$$A = \{ A1, A2, A3, A4, A5, A6, A7 \}$$

$$\Re_{AA} \subseteq A \times A$$

$$\Re_{AA} = \{ (A5, A1), (A7, A5), (A7, A2), (A7, A3), (A6,A3) \}$$

Figura 55 – Exemplo de relação causal entre anomalias

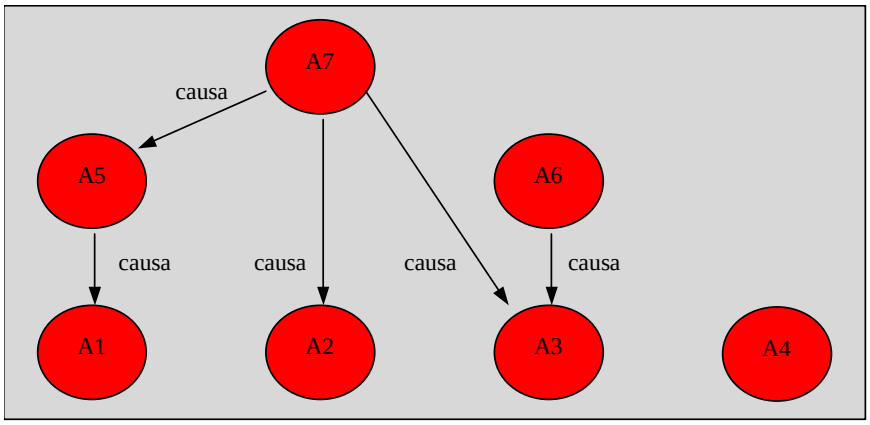


Figura 56 – Exemplo de grafo de relação causal entre anomalias.

É possível notar, por este exemplo, que podem existir anomalias sem relação de causalidade com nenhuma outra anomalia, como é o caso da anomalia A4.

5.4 Relacionamento causal direto entre anomalia e sintoma

Uma anomalia que ocorre em um ambiente distribuído é percebida pelo sistema de diagnóstico através dos sintomas gerados. A ocorrência de uma anomalia no sistema pode causar a observação de nenhum sintoma, um sintoma ou múltiplos sintomas, como ilustrado na Figura 57.

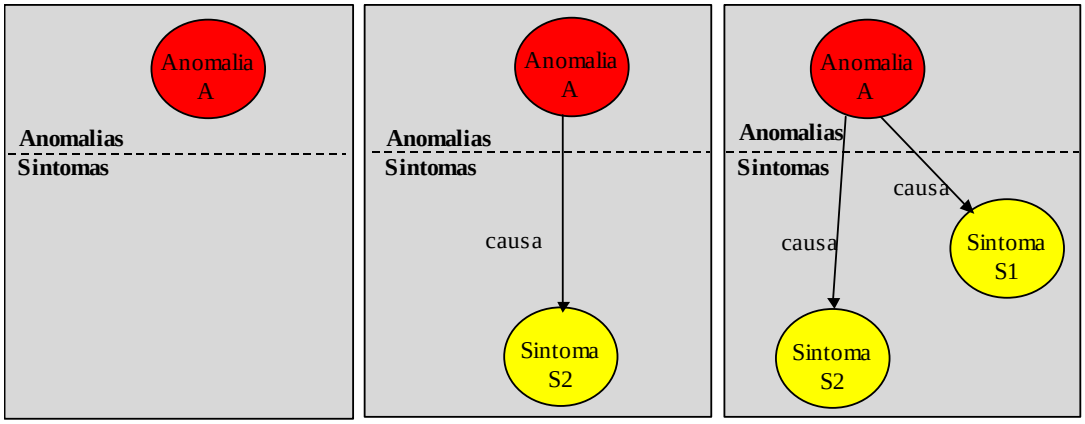


Figura 57 – Exemplos de relacionamento causal entre anomalia e sintoma.

Este relacionamento causal será denominado relacionamento causal direto entre anomalia e sintoma e pode ser representado por $\Re_{AS} \subseteq A \times S$, sendo A o conjunto de anomalias e S, o conjunto de sintomas. A Figura 58 apresenta alguns exemplos de relacionamento causal entre anomalia e sintoma.

```
PROCESSO_TERMINADO    causa SERVIÇO_NÃO_RESPONDE
FALHA_NO_EQUIPAMENTO causa SEM_COMUNICAÇAO_COM_EQUIPAMENTO
ALTA_TEMPERATURA_CPU(anom.) causa ALTA_TEMPERATURA_CPU(sint.)
```

Figura 58 – Exemplo de relações causais entre anomalia e sintoma.

Uma determinada anomalia que ocorre em um componente pode causar a ocorrência de sintomas no mesmo componente (X=Y) ou em um outro componente Y. Além disso, uma anomalia pode gerar um ou mais sintomas, cada um associado ao mesmo componente ou a componentes distintos. Por exemplo, o problema em um enlace físico (componente X) que conecta dois equipamentos Y e Z pode ser observado pelo estado do enlace no equipamento Y e no equipamento Z.

Em um sistema de diagnóstico de um ambiente distribuído as anomalias são observadas sempre através dos sintomas apresentados. Existem situações nas quais uma anomalia causa diretamente um sintoma (caso em que a anomalia possui mapeamento direto em uma observação), sendo simples a tarefa de diagnóstico. Todavia, existem algumas outras situações, como já mencionado anteriormente, nas quais a anomalia não pode ser observada diretamente pelo sistema de diagnóstico devido ao fato de não existir um objeto gerenciado associado que represente o estado do objeto real (componente) causador da anomalia. Apesar disso, a anomalia pode ainda ser identificada de forma indireta, a partir de observações associadas a outras anomalias geradas no sistema, como ilustrado na Figura 59 e Figura 60.

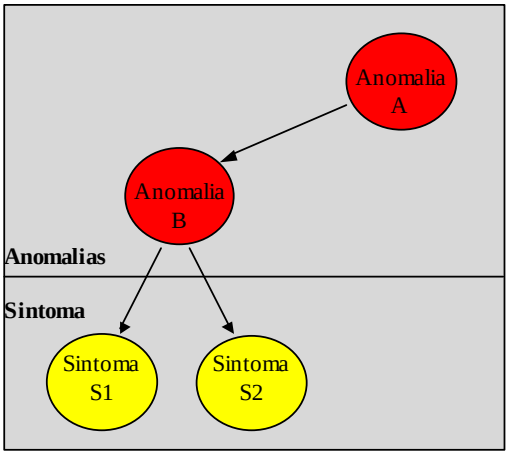


Figura 59 – Exemplo de anomalias sem sintomas diretamente associados.

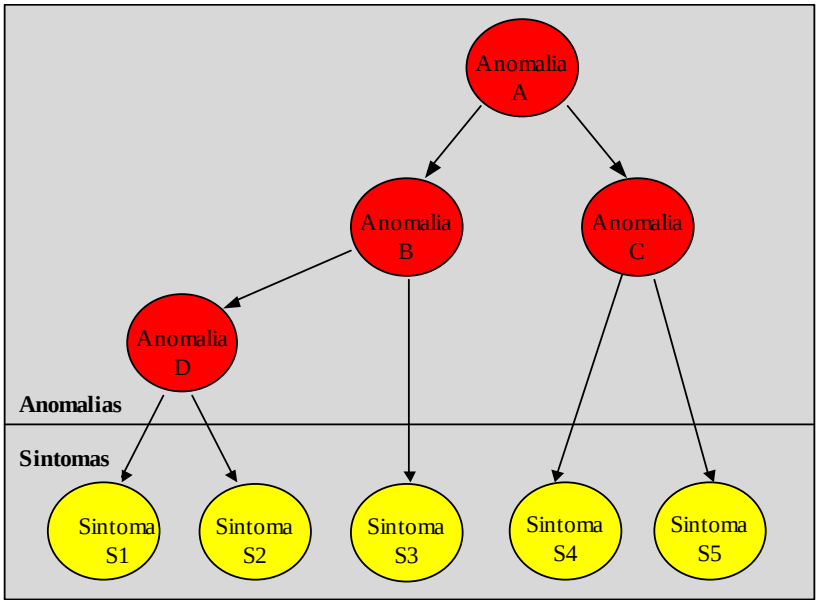


Figura 60 – Exemplo de anomalias sem sintomas diretamente associados.

É possível também ocorrer o caso no qual duas diferentes anomalias podem ser observadas pelo mesmo sintoma, como ilustrado na Figura 61. Neste caso, quando o sintoma S1 é observado, a causa raiz pode ser tanto a anomalia A quanto a anomalia B, dificultando a tarefa de diagnóstico.

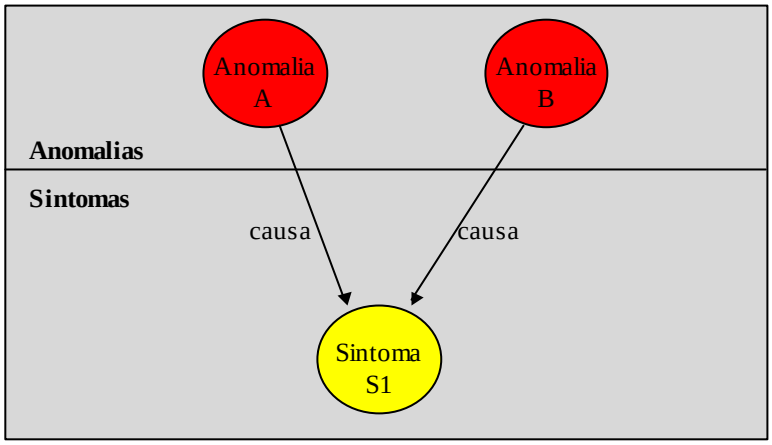


Figura 61 – Diferentes anomalias causando o mesmo sintoma.

5.4.1 Relacionamento causal indireto entre sintomas

Os relacionamentos apresentados anteriormente, relação causal entre anomalias e relação causal entre anomalia e sintoma, definem indiretamente um relacionamento entre sintomas, como ilustrado na Figura 62 e Figura 63.

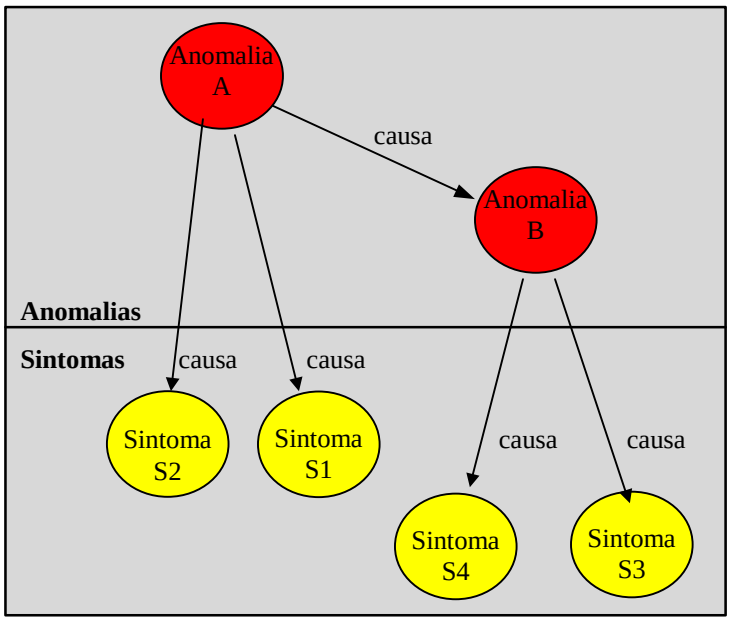


Figura 62 – Diagrama causal entre anomalias e sintomas.

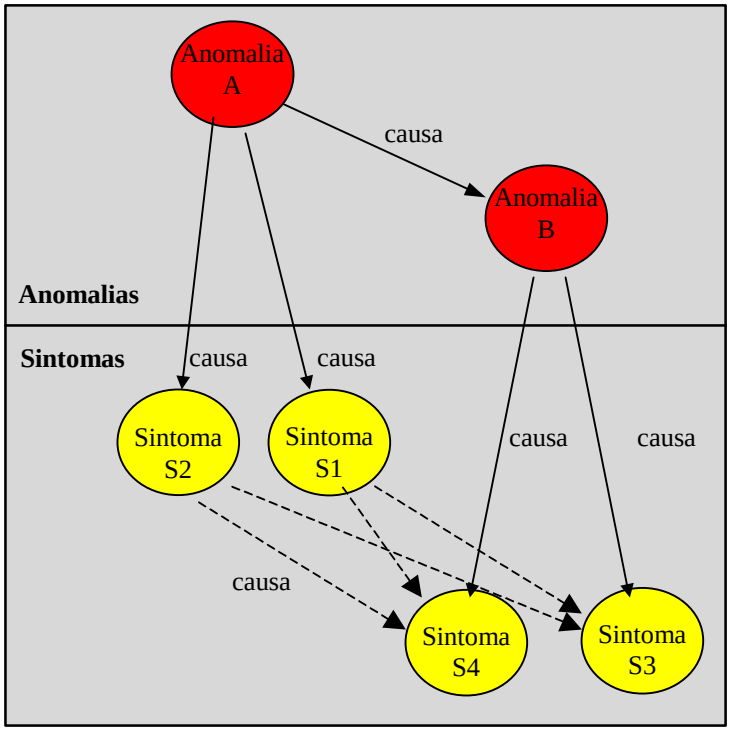


Figura 63 – Relação “causa” entre sintomas.

5.4.2 Relacionamento causal direto entre sintomas

Muitas vezes citados na literatura, o relacionamento causal direto entre sintomas não existe. Um sintoma (observação) não pode causar outro sintoma (observação). O provável é que existam anomalias intermediárias que não foram modeladas. Este é o caso do grafo causal apresentado na Figura 64 extraído de Kinger (1995) e do grafo causal apresentado em Lemos (1999).

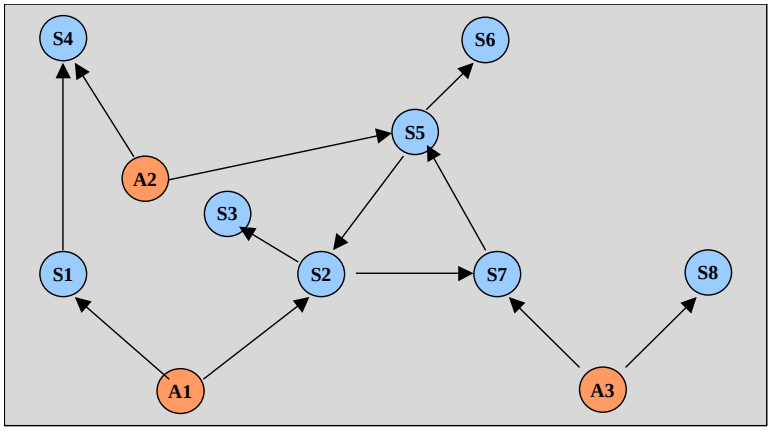


Figura 64 – Exemplo de grafo causal, adaptado de Kinger (1995)

5.5 Conclusão

Neste capítulo foram apresentadas várias definições importantes para a seções subseqüentes. Particularmente, foi criada definição, o objeto intermediário, que é um componente fundamental para o modelamento das observações.

Este capítulo também se tratou do relacionamento entre anomalias e sintomas, distinguindo-as explicitamente no modelo, o que não é comum na literatura. Esta forma de modelamento de anomalia e sintoma possibilita explicitar algumas relações causais que geralmente são subtendidas.

6. Proposta de modelagem da imprecisão temporal das observações

A concepção de um sistema de diagnóstico voltado para a análise de determinados sistemas, particularmente sistemas apoiados sobre ambiente computacional distribuído (como diagnóstico de rede de comunicação ou de sistemas distribuídos), deve considerar a forma pela qual as observações são obtidas. Estas observações podem se apresentar defasadas no tempo devido a diversos fatores. Além disso, a tentativa de diagnóstico no momento imediato à chegada de um sintoma esbarra na falta de informações (observações) atualizadas e incertezas temporais decorrentes de diversas naturezas, denominados neste trabalho de "problemas temporais de uma observação".

A seguir, é apresentada uma seção a respeito da caracterização temporal da observação e são descritos os "problemas temporais de uma observação". Em seguida é apresentada uma técnica de modelamento da observação de forma a permitir que estas imprecisões fiquem explícitas ao sistema de diagnóstico. Este modelamento envolve a definição dos seguintes intervalos:

- intervalo de possibilidade (decorrente de imprecisões temporais a respeito do instante de início e término da ocorrência de um determinado estado de um objeto gerenciado);
- intervalo de certeza: (decorrente de imprecisões temporais a respeito do instante de início e término da ocorrência de um determinado estado de um objeto gerenciado);
- intervalo de incerteza (decorrente de incertezas geradas pelo não conhecimento do estado atual do objeto gerenciado).

O acréscimo destas informações temporais pode possibilitar uma maior precisão ao sistema de diagnóstico, evitando falsos positivos e falsos negativos. Também propicia definir um grau de confiança para cada resultado de diagnóstico apresentado.

6.1 Caracterização temporal da observação quanto à sua defasagem

As observações recebidas por um sistema de diagnóstico, a respeito do ambiente gerenciado, podem ser classificadas quanto à defasagem no tempo (entre sua ocorrência e sua observação) em três classes:

- imediata;
- defasada em até 1 ciclo;
- defasada em até 2 ciclos.

Uma observação imediata é aquela geralmente derivada de notificações assíncronas[20] enviadas por agentes de gerenciamento. É importante frisar que, mesmo neste caso, o agente de monitoração pode utilizar internamente uma técnica de amostragem periódica para atualização do estado do objeto gerenciado. Como tais agentes geralmente são responsáveis pela monitoração de objetos gerenciados locais, seus períodos de amostragem são pequenos e, neste caso, podem ser considerados como observações imediatas.

Uma observação defasada geralmente é aquela decorrente da monitoração realizada por plataformas de gerenciamento ou similares (que fazem o papel de gerentes de gerenciamento). Estes gerentes em geral utilizam monitoração baseada na técnica de amostragem remota periódica[21].

Devido ao fato de a monitoração ser geralmente "*in-band*", ou seja, consumindo banda da própria infra-estrutura de comunicação, e ser realizada sobre centenas ou milhares de equipamentos, o período de amostragem não pode ser muito pequeno

---

[20] O Anexo 1 apresenta maiores detalhes.

[21] Além da amostragem remota periódica, a observação defasada também é decorrente da monitoração através de agentes que se utilizam de MIBs que possuem funcionalidade de amostragem periódica ou MIBs programáveis, como é o caso das seguintes MIBs: RMON1 MIB [RFC1757 1995], RMON2 MIB [RFC2021 1997], SMON MIB [RFC2613 1999], "Distributed Management MIB" [RFC2925 2000], "Event MIB" [RFC2981 2000] e "Script MIB" [RFC2593].

para não causar consumo excessivo de banda de comunicação ou mesmo sobrecarga de processamento do próprio gerente. Também, o gerenciamento realizado em médias e grandes corporações dispersas geograficamente pode utilizar enlaces de baixa capacidade. Oliveira (1998) relata que corporações brasileiras dispersas geograficamente utilizam em larga escala enlaces de baixa capacidade, impondo restrições extremamente fortes ao gerenciamento.

Nos exemplos apresentados a seguir será utilizado o caso de um estado de um objeto gerenciado (representando, por exemplo, uma anomalia) e a observação resultante desse estado pelo sistema de diagnóstico, considerando-se somente dois estados: NORMAL e ANÔMALO. O estado anômalo, para o sistema de diagnóstico, é um sintoma.

Em alguns casos pode não ser suficiente mapear o comportamento de um objeto gerenciado em somente dois estados (como por exemplo NORMAL e ANÔMALO), sendo necessária a utilização de múltiplos estados como, por exemplo:

- taxa de utilização: BAIXA, NORMAL, ALTA, SATURADA;
- taxa de ocupação: NORMAL, QUASE_CHEIA, CHEIA.

O motivo desta escolha é simplificar o exemplo e, conseqüentemente, facilitar a compreensão do tópico.

Neste trabalho, em relação à defasagem, foram identificadas as seguintes classes de observações:

- observação não defasada;
- observação defasada em até 1 ciclo;
- observação defasada em até 2 ciclos.

6.1.1 Observação não defasada

Uma observação não defasada é geralmente aquela resultante de objetos gerenciados cujo estado é conhecido através do recebimento de eventos assíncronos[22], como

[22] Para maiores informações deve ser consultado o Anexo 1.

SNMP-TRAP no gerenciamento SNMP (RFC1215, 1991) ou notificações no modelo de referência OSI, gerados sempre quando ocorre uma mudança de estado.

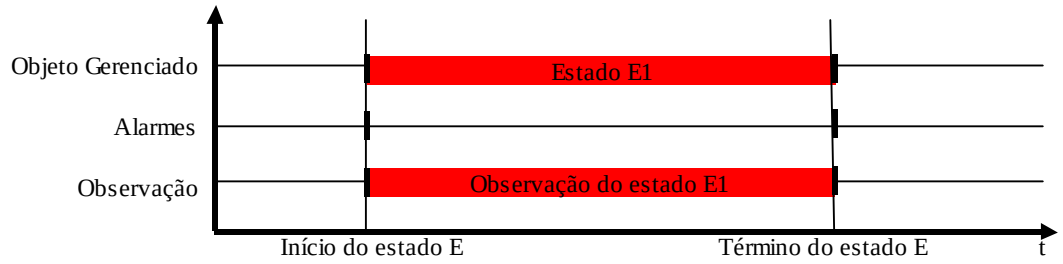


Figura 65 – Exemplo de observação não defasada

Estes eventos são enviados imediatamente pelo agente de gerenciamento no momento em que detecta a mudança de estado do objeto gerenciado (por exemplo, passa do estado NORMAL para o ANÔMALO ou vice versa). A Figura 65 mostra um exemplo de observação não defasada.

Não estão incluídas nesta classe as observações decorrentes de notificações que não informem imediatamente quando ocorre uma mudança de estado.

6.1.2 Observação defasada em até 1 ciclo

Uma observação defasada em até 1 ciclo é geralmente aquela resultante da amostragem periódica de objetos gerenciados cuja expressão para determinação de seu estado envolve somente valores de objetos gerenciados da última amostragem.

```
Estado_operacional_interface_ethernet = ifOperState
```

Figura 66 – Exemplo de expressão de uma observação defasada em até 1 ciclo.

A técnica de amostragem causa uma defasagem entre o intervalo de ocorrência de um estado $E=<T_i(E),T_f(E)>$ e o intervalo de estado observado $O(E)=<T_i(O(E)),T_f(O(E))>$. Esta defasagem pode ser de até 1 ciclo de amostragem e pode ser expressa pelas seguintes expressões:

$$T_i(E) < T_i(O(E)) \text{ e } T_i(O(E)) - T_i(E) < P$$
$$T_f(E) < T_f(O(E)) \text{ e } T_f(O(E)) - T_f(E) < P$$

Para ilustrar, considere-se o exemplo, mostrado na Figura 67, de um objeto gerenciado cuja observação pode estar defasada em até 1 ciclo. Nele, é apresentada a ocorrência de um determinado estado "E" entre os instantes $T_i(E)$ e $T_f(E)$, sendo O(E) a representação da observação do estado "E". O objeto gerenciado (ou objetos gerenciados) é observado em intervalos periódicos, com período P. Na figura, $T_i(E)$ representa o instante inicial do estado "E" (o instante de transição de estado para "E"), $T_f(E)$ o instante final do estado A, $T_i(O(E))$ o instante inicial da observação do estado "E" e $T_f(O(E))$ o instante final da observação do estado "E".

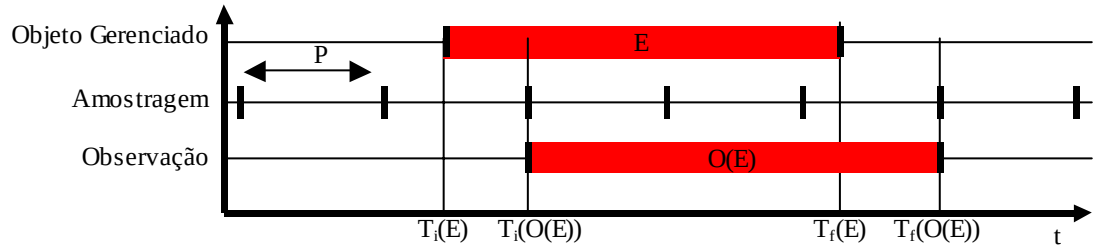


Figura 67 - Exemplo de observação defasada em até 1 ciclo

6.1.3 Observação defasada em até 2 ciclos

Uma observação defasada em até 2 ciclos é geralmente aquela resultante de uma expressão numérica envolvendo o resultado da amostragem corrente e da anterior sobre um ou mais objetos gerenciados. A Figura 68 mostra o exemplo de uma expressão de observação defasada em até 2 ciclos.

```
Ethernet_half_duplex_port_utilization =
      ((ΔifInOctets + ΔifOutOctets) * 8) / (Δt * ifSpeed)
   Sendo:
      ΔifInOctets  = ifInOctets(t2)  – ifInOctets(t1)
      ΔifOutOctets = ifOutOctets(t2) – ifOutOctets(t1)
      Δt           = t2 – t2
```

Figura 68 – Exemplo de expressão de uma observação defasada em até 2 ciclos

Neste caso existem duas situações possíveis:

- situação A: a primeira amostragem realizada após o início do estado “E” causa impacto suficiente no resultado da expressão numérica, indicando uma mudança de estado;
- situação B: a primeira amostragem realizada após o início do estado “E” não causa impacto suficiente no resultado da expressão numérica, não indicando a mudança de estado. A mudança de estado será detectada somente no instante da próxima amostragem.

6.1.3.1 Situação A

A amostragem realizada imediatamente após o início do estado “E” ($T_i(E)$) causou impacto suficiente no resultado da expressão numérica, indicando uma mudança de estado.

Seja o exemplo mostrado na Figura 69 idêntico ao anterior, exceto por a observação poder estar defasada em até 2 ciclos. Apesar disso, a primeira amostragem realizada após o início do estado “E” causa impacto suficiente no resultado da expressão, indicando a mudança de estado

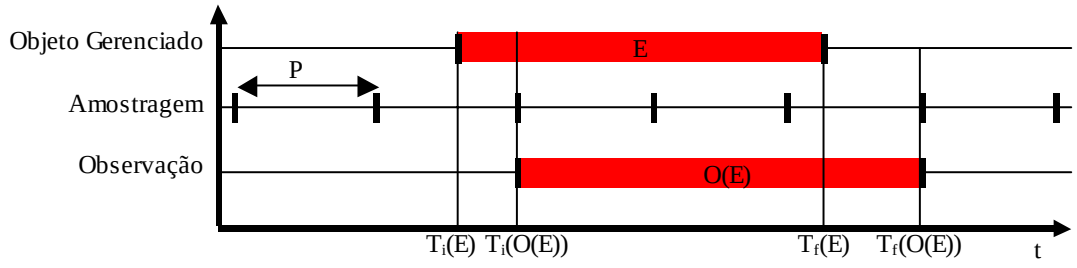


Figura 69 - Exemplo de observação defasada em até 2 ciclos que apresenta defasagem de até 1 ciclo

Neste caso a observação está defasada em no máximo 1 ciclo, ou seja:

$$T_i(E) < T_i(O(E)) \text{ e } T_i(O(E)) - T_i(E) < P$$
$$T_f(E) < T_f(O(E)) \text{ e } T_f(O(E)) - T_f(E) < P,$$

6.1.3.2 Situação B

A amostragem realizada imediatamente após o início da anomalia A ($T_i(E)$) não causou impacto suficiente no resultado da expressão numérica, não indicando ainda

uma mudança de estado. Esta será percebida somente no instante da próxima amostragem. O exemplo mostrado na Figura 70 ilustra esta situação.

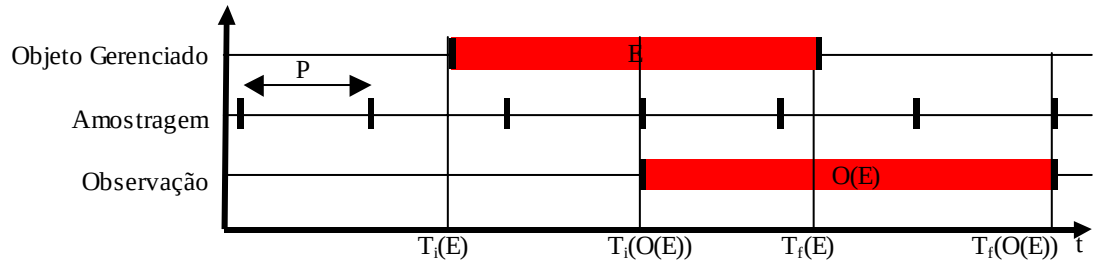


Figura 70 – Exemplo de uma observação defasada em até 2 ciclos.

Neste caso o sintoma está defasado em quase 2 ciclos. Compondo com o resultado da situação A, esta defasem pode ser expressa pelas seguintes expressões:

$$T_i(E) < T_i(O(E)) \text{ e } T_i(O(E)) - T_i(E) < 2P$$
$$T_f(E) < T_f(O(E)) \text{ e } T_f(O(E)) - T_f(E) < 2P$$

6.2 Problemas temporais de uma observação

Os principais problemas associados ao tempo de uma observação, denominados aqui "problemas temporais de uma observação", são:

- defasagem de tempo na observação do estado de um objeto gerenciado (relacionamento observação-objeto gerenciado);
- estado atual do objeto gerenciado disponível somente na próxima observação (relacionamento observação-objeto gerenciado);
- relacionamento de observações defasadas no tempo (relacionamento entre observações);

6.2.1 Defasagem de tempo na observação do estado de um objeto gerenciado

O problema de defasagem de tempo entre o estado do objeto gerenciado e o estado observado ocorre principalmente nas observações derivadas de amostragens periódicas cujo problema foi descrito na seção 6.1:

- observações defasadas em até 1 ciclo (seção 6.1.2);
- observações defasadas em até 2 ciclos (seção 6.1.3).

6.2.2 Estado atual do objeto gerenciável disponível somente na próxima observação

Para cada observação decorrente de amostragem existe um intervalo de tempo, que se inicia do instante da última amostragem até o instante corrente, no qual não existe informação atualizada sobre o objeto gerenciado. Qualquer alteração de estado do objeto gerenciado só será conhecida na próxima amostragem.

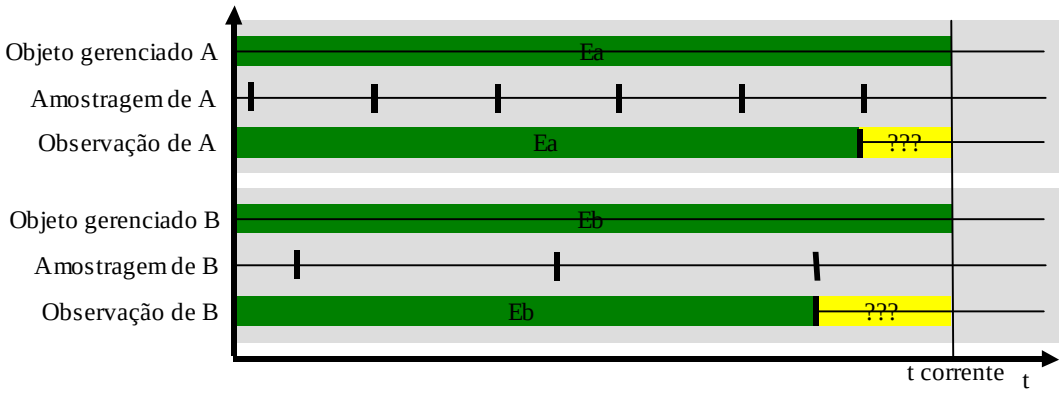


Figura 71 – Exemplo de intervalos no qual não existe informação atualizada sobre o objeto gerenciado

A Figura 71 mostra a observação de dois objetos gerenciados A e B, explicitando os intervalos nos quais o estado atual do objeto é desconhecido.

6.2.3 Relacionamento de observações defasadas no tempo

Outro problema apresentado está relacionado ao relacionamento das observações. Este relacionamento pode não ser trivial pois as observações podem estar defasadas. Esta defasagem pode ser causada pela utilização de diferentes períodos ou fases de amostragem.

6.2.3.1 Observações defasadas devido à utilização de períodos de amostragem diferentes

Um monitor, para realizar a observação periódica de um objeto gerenciado pode utilizar diferentes períodos de amostragem, definidos de acordo com:

- a freqüência de alteração de seu estado;
- sua importância no ambiente.

Por exemplo, o período de amostragem dos objetos gerenciados associados à observação do estado de um enlace de comunicação de um roteador pode ser de 5 minutos enquanto que o período de amostragem da taxa de utilização de um sistema de arquivos pode ser de 1 hora, já que a variação da taxa de utilização usualmente não se altera muito no decorrer do tempo.

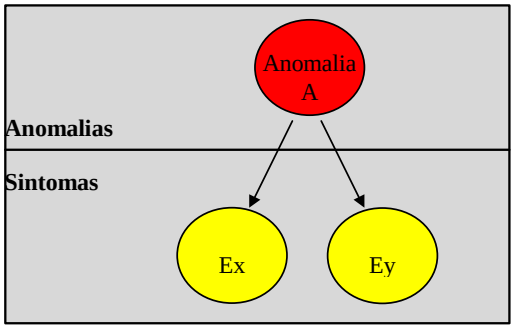


Figura 72 – Exemplo de relação causal.

Para ilustrar esta situação, suponha-se uma anomalia A que cause impacto sobre o estado dos objetos gerenciados X e Y (relacionamento causal imediato), fazendo com que apresentem imediatamente os sintomas Ex e Ey, respectivamente. As observações dos estados Ex e Ey são respectivamente O(Ex) e O(Ey), sendo ambas observações defasadas em até 1 ciclo. Os períodos de amostragem são Px e Py respectivamente, sendo Px diferente de Py. A Figura 73 mostra o intervalo no qual ocorreram os estados Ex e Ey e as respectivas observações O(Ex) e O(Ey).

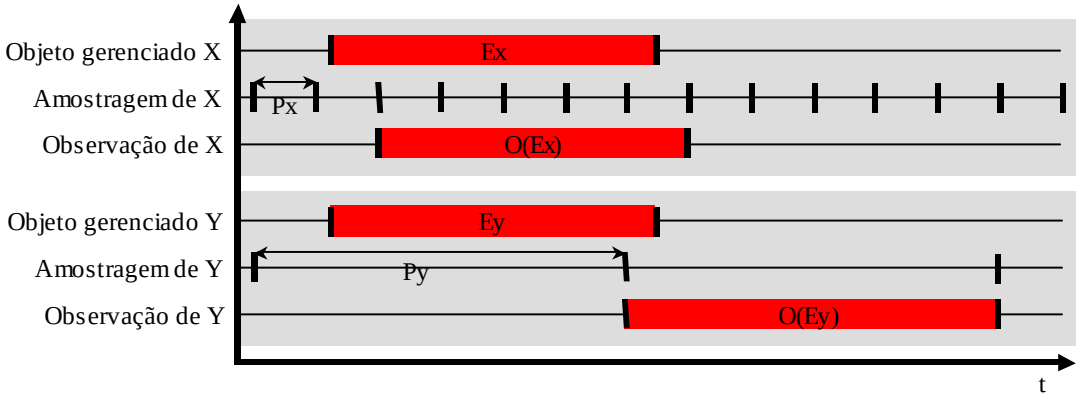


Figura 73 – Exemplo de observações que utilizam períodos diferentes de amostragem

É possível observar que, devido à diferença de período de amostragem, não é trivial perceber o relacionamento destes sintomas (O(Ex) e O(Ey)) por possuírem apenas um pequeno intervalo de intersecção, como mostrado na Figura 74.

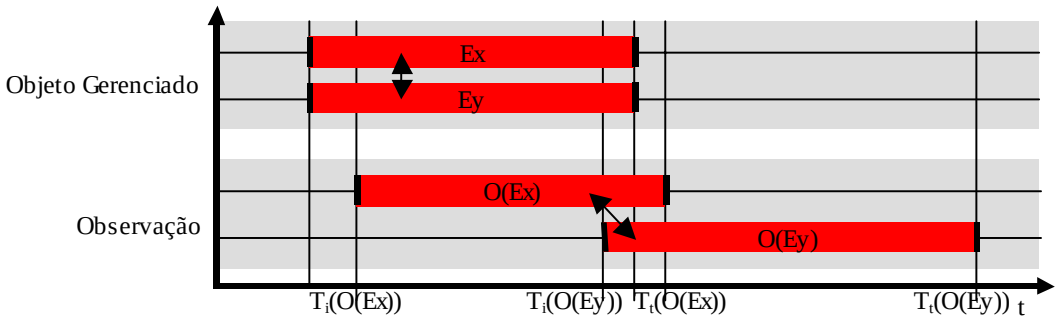


Figura 74 – Exemplo do impacto da defasagem de observações no relacionamento causal devido à utilização de diferentes períodos de amostragem

Sistemas de correlação possuem sérias restrições neste tipo de situação. Sistemas de diagnóstico também têm dificuldade de tratar tal situação. Particularmente, os que são capazes de prever tal situação são:

- os “sistemas de diagnóstico atemporal sobre coleção sintomas”[23]. Apesar de possibilitar este relacionamento esta classe de sistemas de diagnóstico apresenta vários falsos positivos.
- os “sistemas de diagnóstico temporal variante no tempo”[24]. São sistemas mais complexos, porém muito mais adequados para esta situação.

[23] O sistema de diagnóstico atemporal sobre coleção de sintomas está descrito na seção 3.1.2.

6.2.3.2 Observações defasadas devido à utilização de fases de amostragem diferentes

Mesmo nos sistemas nos quais é definido um mesmo período de amostragem, a diferença de fase entre as amostragens pode causar o mesmo problema descrito anteriormente. A Figura 75 ilustra uma destas situações.

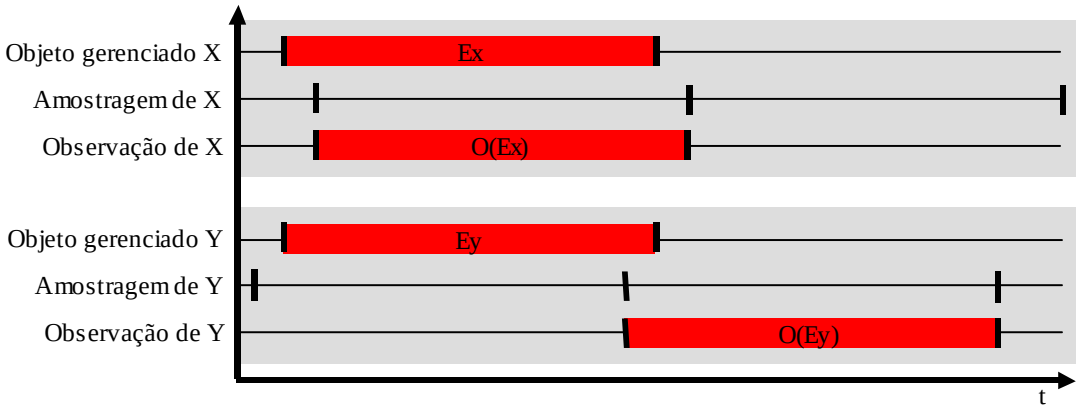


Figura 75 - Exemplo de observações que utilizam fases diferentes de amostragem.

Novamente, é possível observar que, devido à diferença de fase de amostragem, o relacionamento destes sintomas (O(Ex) e O(Ey)) não é trivial devido ao fato de as observações possuírem apenas um pequeno intervalo de intersecção, como mostrado na Figura 76.

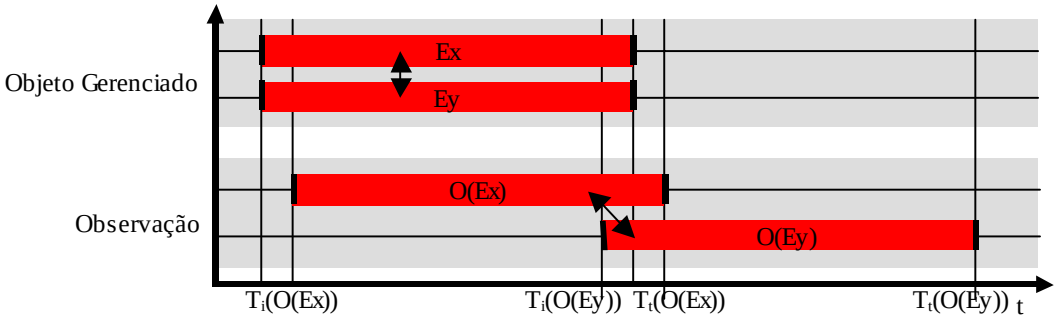


Figura 76 - Exemplo do impacto da defasagem de observações no relacionamento causal devido à utilização de diferentes fases de amostragem

[24] O sistema de diagnóstico temporal variante no tempo está descrito na seção 3.1.5.

6.3 Modelamento das incertezas temporais associadas às observações

Como mostrado anteriormente, a observação que é utilizada por um sistema de diagnóstico não é precisa em relação ao instante de ocorrência da mudança de estado. A utilização da observação nesta forma bruta pode levar a erros quando manipulada por um sistema de correlação ou de diagnóstico.

Contudo, existem informações que são descartadas e, se utilizadas convenientemente, poderiam auxiliar em uma definição mais aproximada dos momentos de mudança de estado dos objetos gerenciados.

Este trabalho propõe um novo modelo de observação que incorpora informações a respeito dos possíveis momentos de mudança de estado e, também, trata as situações nas quais existe ausência de observação (falhas na amostragem ou desconhecimento do estado do objeto próximo ao momento corrente), que é igualmente importante explicitar.

Isto permitiria ao sistema de diagnóstico contar com uma observação mais sofisticada. A geração desta nova observação pode ser realizada por um sistema intermediário que possua o controle ou conhecimento do processo de monitoramento, principalmente do período de amostragem.

6.3.1 A nova representação da observação

Na nova forma de representação da observação, ela é um conjunto de intervalos que têm como finalidade representar o comportamento do objeto gerenciado através da evolução de seus estados no tempo, incorporando as incertezas temporais existentes.

A nova representação da observação utiliza os seguintes tipos de intervalos:

- IPI(E) -Intervalo de Possibilidade de Início (de estado de objeto gerenciado);
- IC(E) - Intervalo de Certeza (do estado de objeto gerenciado);
- IPT(E) - Intervalo de Possibilidade de Término (de estado de objeto gerenciado);

- II - Intervalo de Incerteza (em relação ao estado de objeto gerenciado).

A Figura 77 mostra um exemplo de modelamento da observação associada ao intervalo entre início e término da ocorrência do estado E em um objeto gerenciado.

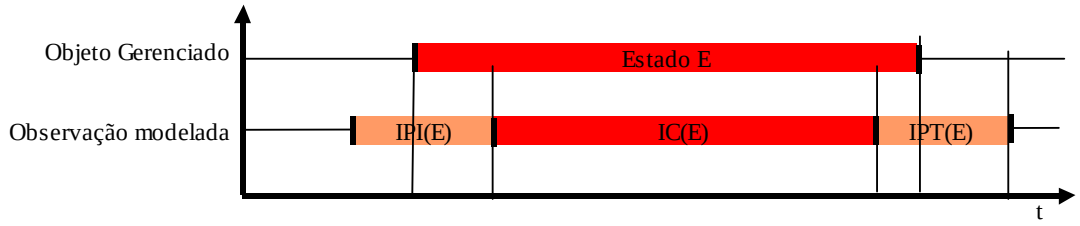


Figura 77 – Exemplo de modelamento da observação do estado E de um objeto gerenciado

Muitas vezes o sistema de diagnóstico necessita conhecer não somente os estados anômalos (sintomas), mas também o restante do conjunto de estados associados a um objeto gerenciado. A Figura 78 ilustra como é modelado o comportamento de um objeto gerenciado em relação à completude de seus estados, neste caso estados X e Y, considerando também os intervalos de incerteza.

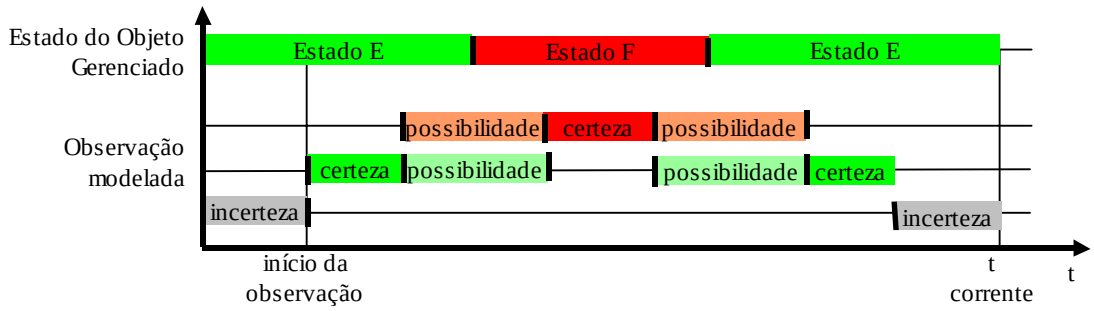


Figura 78 – Exemplo de modelamento da observação de todos os estados de um objeto gerenciado

A seguir, serão apresentadas as definições a respeito destes intervalos. As seções a seguir descrevem com maior detalhamento o significado e a importância destes intervalos para um sistema de diagnóstico temporal.

Definição 15: IO(E) - Intervalo de ocorrência de estado

O intervalo de ocorrência de um estado E em um objeto gerenciado, denotado por IO(E), é definido com sendo o intervalo de tempo entre o início real da ocorrência do estado e o término da ocorrência do estado.

Definição 16: IC(E) - Intervalo de certeza de ocorrência de estado

O intervalo de certeza da ocorrência de um estado E em um objeto gerenciado, denotado por IC(E), representa o intervalo de tempo no qual se tem certeza da ocorrência deste estado E.

Definição 17: IPI(E) - Intervalo de possibilidade de início de ocorrência de estado

O intervalo de possibilidade de início de um estado E em um objeto gerenciado, denotado por IPI(E), representa um intervalo de tempo que contém o instante de transição para o estado E.

Definição 18: IPT(E) - Intervalo de possibilidade de término de ocorrência de um estado

O intervalo de possibilidade de término de um estado E em um objeto gerenciado, denotado por IPT(E), representa um intervalo de tempo que contém o instante de transição para outro estado diferente de E.

6.3.2 O modelo tradicional de geração de observações

O modelo tradicional de monitoramento utilizado pelos sistemas de correlação e sistemas de diagnóstico está ilustrado na Figura 79.

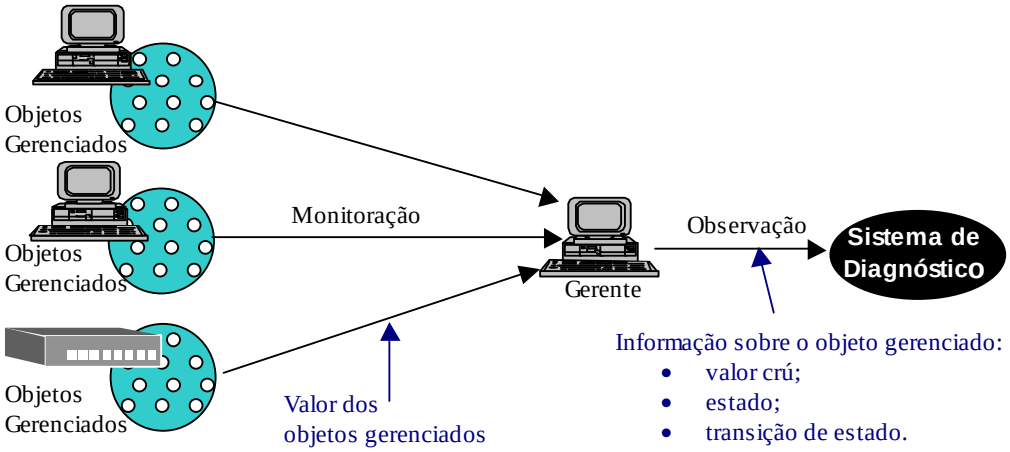


Figura 79 – Modelo tradicional de monitoramento utilizado por um sistema de diagnóstico.

A Figura 80 mostra as entidades envolvidas no modelo tradicional de geração de observações.

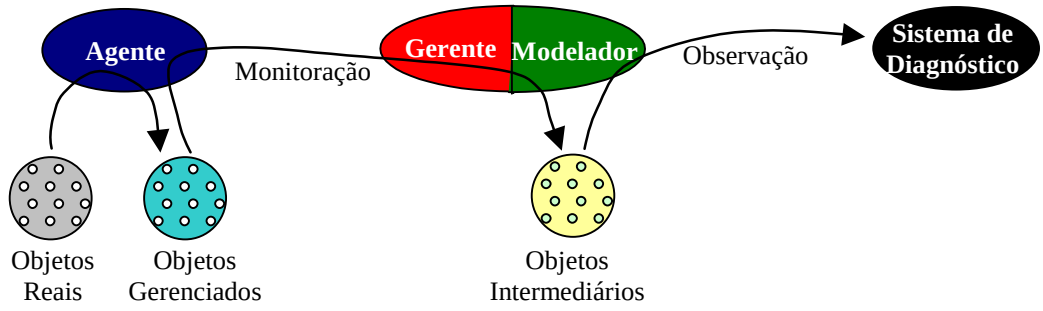


Figura 80 – Entidades envolvidas no modelo tradicional de geração de observações para um sistema de diagnóstico.

6.3.3 O novo modelo para geração de observações

O novo modelo é uma variante do modelo tradicional, pelo acréscimo de uma entidade: o **Modelador**. O papel do Modelador é armazenar informações de contexto associadas à monitoração dos objetos gerenciados de forma a possibilitar a representação do estado dos objetos gerenciados através dos intervalos descritos anteriormente. Assim, surge uma nova visão associada ao objeto gerenciado que é chamada de “objeto intermediário”. Este novo modelo está ilustrado na Figura 81.

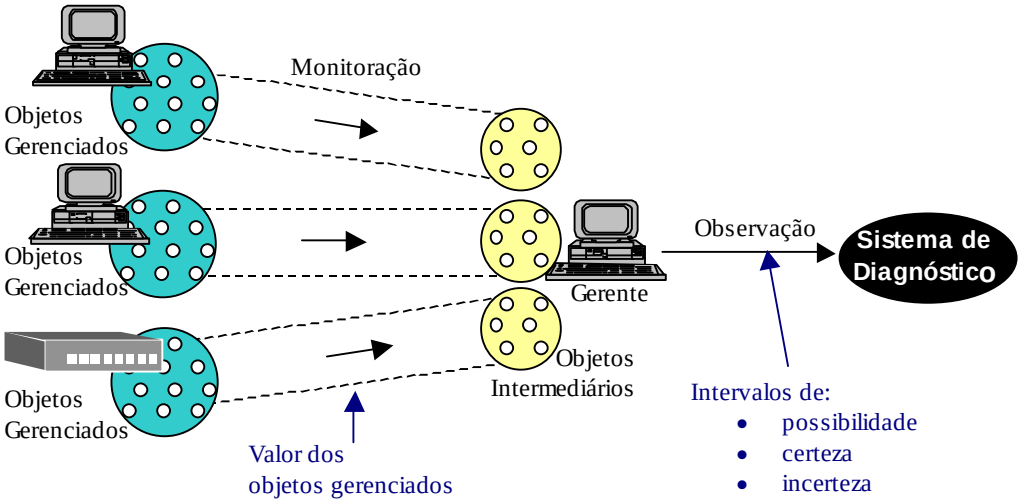


Figura 81 – Novo modelo de monitoramento utilizado por um sistema de diagnóstico.

A Figura 82 mostra as entidades envolvidas no modelo temporal de geração de observações.

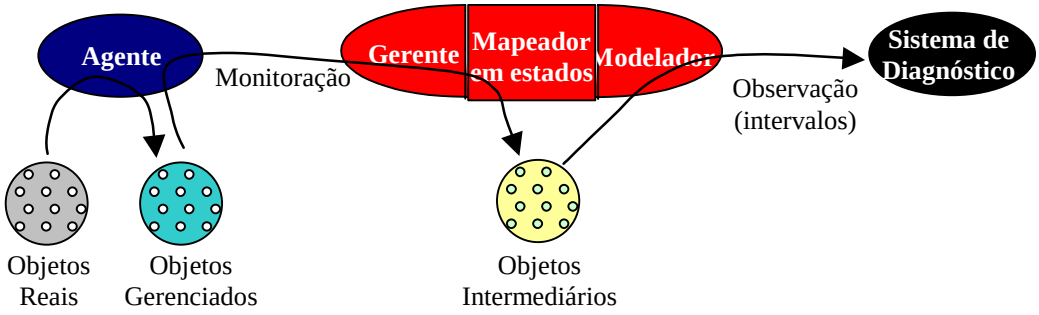


Figura 82 - Entidades envolvidas no novo modelo de geração de observações para um sistema de diagnóstico.

6.4 Modelamento da observação: Intervalos de possibilidade e certeza

Os intervalos de possibilidade e certeza da ocorrência de um determinado estado têm como objetivo caracterizar, da maneira mais próxima possível, os instantes de início e término deste estado, se necessário explicitando as imprecisões temporais geradas no processo de amostragem.

6.4.1 Modelamento de observações não defasadas

Os intervalos de possibilidade e certeza são importantes para o modelamento da imprecisão temporal decorrente das amostragens. Como as observações não defasadas são geralmente as decorrentes de eventos assíncronos, elas não possuem defasagem decorrente de amostragem e portanto não apresentam estados de possibilidade, como ilustrado na Figura 83.

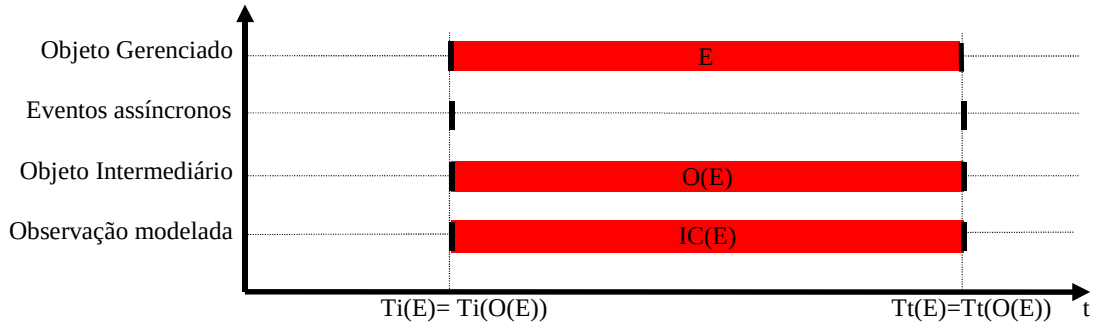


Figura 83 – Exemplo de modelamento de uma observação não defasada.

Assim, supondo a ocorrência da observação O(E) associada ao estado E de um objeto gerenciado entre os instantes $T_i(O(E))$ e $T_f(O(E))$, o intervalo de certeza da ocorrência do estado E (IC(E)) pode ser definido pela seguinte expressão:

$$IC(E) = < T_i(O(E)) , T_i(O(E)) >$$

6.4.2 Modelamento de observações defasadas em até 1 ciclo

Se o intervalo de amostragem associado à monitoração de um objeto gerenciado for conhecido é possível definir os intervalos de possibilidade e certeza para seus estados. A Figura 84 mostra um exemplo desta situação.

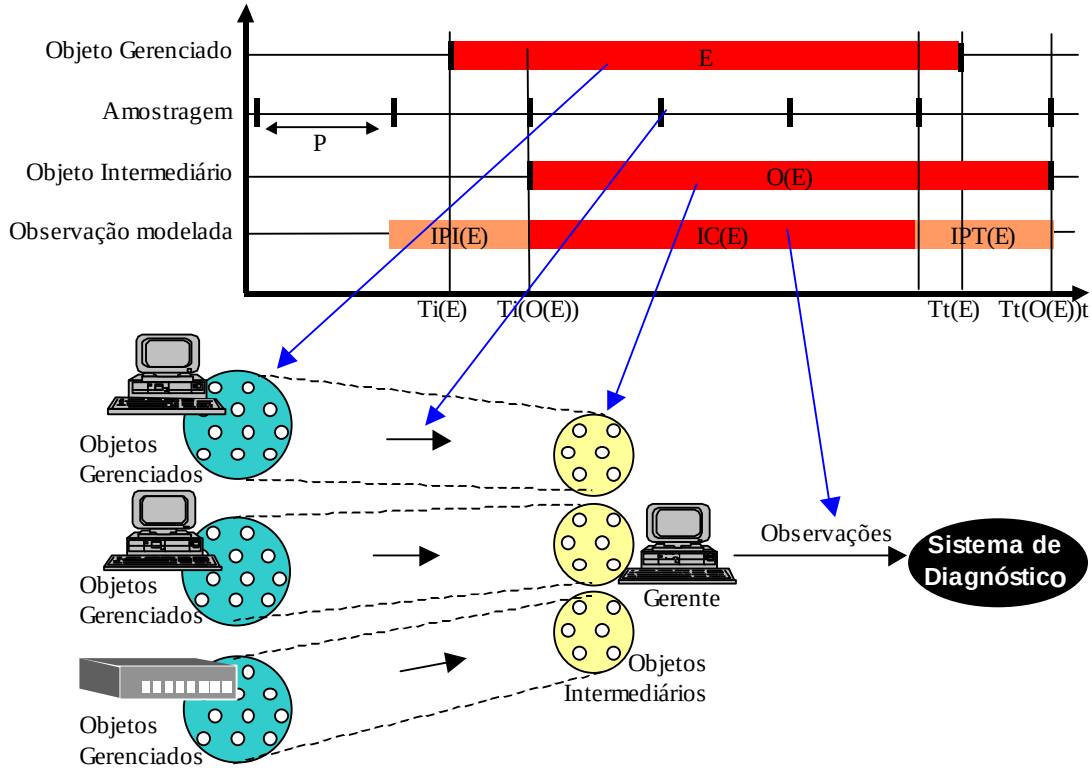


Figura 84 - Exemplo de modelamento de uma observação defasada em até 1 ciclo.

Assim, seja a monitoração de um objeto gerenciado com período de amostragem P que pode estar defasado em até 1 ciclo, uma ocorrência de estado “E” no objeto gerenciado e sua respectiva observação O(E) ocorrida entre os instantes $T_i(O(E))$ e $T_f(O(E))$. Os intervalos de possibilidade de início (IPI(E)), de certeza (IC(E)) e de possibilidade de término (IPT(E)) do estado “E” podem ser definidos pelas seguintes expressões:

$$IPI(E) = < T_i(O(E)) - P \ , \ T_i(O(E)) >$$
$$IC(E) = < T_i(O(E)) \ , \ T_t(O(E)) - P >$$
$$IPT(E) = < T_t(O(E)) - P \ , \ T_t(O(E)) >$$

6.4.3 Modelamento de observações defasadas em até 2 ciclos

A Figura 85 mostra um exemplo de observação defasada em até 2 ciclos modelada de maneira análoga.

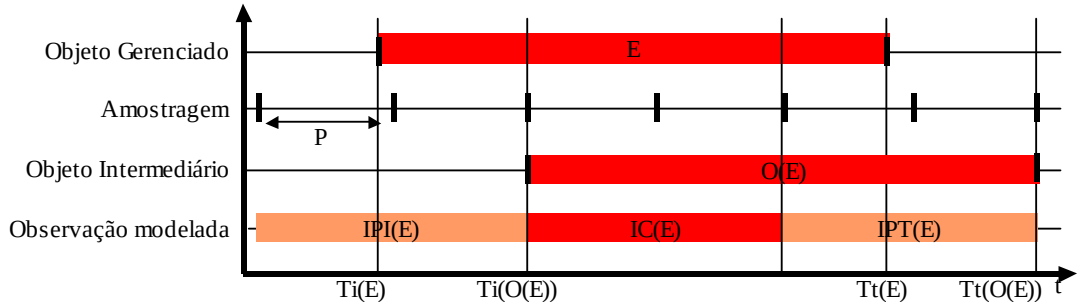


Figura 85 - Exemplo de modelamento de uma observação defasada em até 2 ciclos

Assim, seja a monitoração de um objeto gerenciado com período de amostragem P que pode estar defasado em até 2 ciclos, uma ocorrência de estado “E” no objeto gerenciado e sua respectiva observação O(E) ocorrida entre os instantes $T_i(O(E))$ e $T_f(O(E))$. Os intervalos de possibilidade de início (IPI(E)), de certeza (IC(E)) e de possibilidade de término (IPT(E)) do estado “E” podem ser definidos pelas seguintes expressões:

$$IPI(A) = < T_i(O(E)) - 2P \ , \ T_i(O(E)) >$$
$$IC(A) = < T_i(O(E)) \ , \ T_t(O(E)) - 2P >$$
$$IPT(A) = < T_t(O(E)) - 2P \ , \ T_t(O(E)) >$$

6.4.4 Trabalhando com a completude do conjunto de estados

O sistema de diagnóstico pode trabalhar somente com os sintomas (estados anômalos) de cada objeto intermediário ou com a completude de seus estados. Tais estados são modelados exatamente da mesma forma.

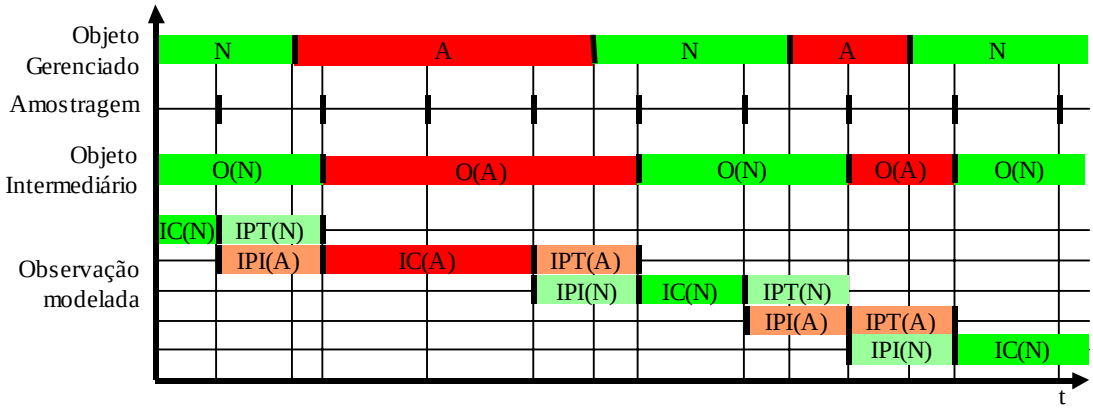


Figura 86 - Exemplo de modelamento de uma observação defasada em até 1 ciclo, com a completude de seus estados.

O exemplo da Figura 86 mostra que o modelamento do estado de um objeto intermediário possui somente dois estados: N=NORMAL e A=ANÔMALO e cuja observação pode estar defasada em até 1 ciclo. A Figura 87 mostra o mesmo exemplo para observação defasada em até 2 ciclos.

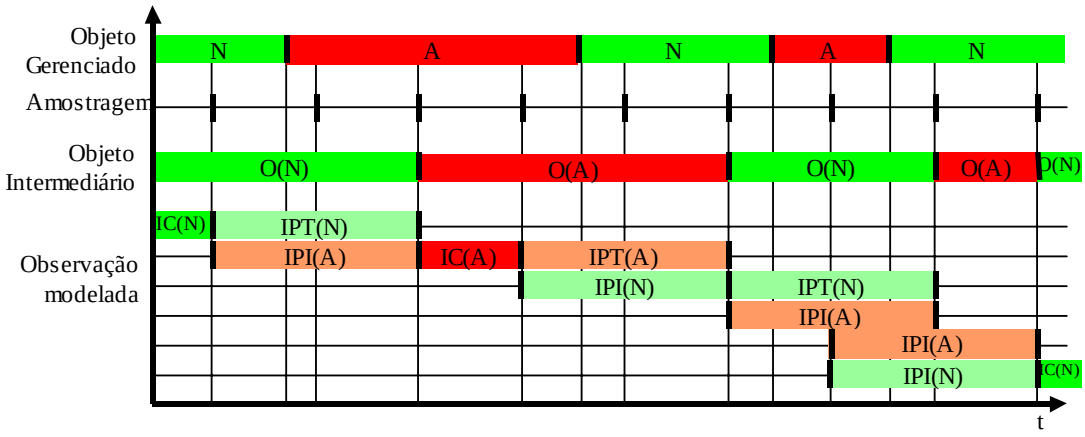


Figura 87 - Exemplo de modelamento de uma observação, defasada em até 2 ciclos, com a completude de seus estados.

Esta forma de modelamento vale também para qualquer objeto intermediário que possua mais que dois estados.

6.5 Modelamento da observação: intervalos de incerteza

Os intervalos de incerteza podem ocorrer tanto na monitoração de objetos gerenciados baseados em amostragens periódicas quanto nos baseados em alarmes assíncronos, e podem ser causados principalmente por:

- falha na monitoração (perda de eventos assíncronos ou perda de amostragens);
- incerteza próximo ao instante corrente devido ao intervalo de amostragem;
- início do processo de monitoração.

6.5.1 Intervalo de incerteza gerado pela perda de observação

6.5.1.1 Perda de eventos assíncronos

A perda de eventos assíncronos é um sério problema em sistemas de monitoração, de correlação e de diagnóstico, principalmente quando é utilizado um protocolo de gerenciamento com entrega não confiável, como é o caso do SNMP sobre UDP.

Geralmente, não é possível determinar quando ocorre uma perda. Nas situações onde for possível, este período pode ser modelado como um intervalo de incerteza (incerteza a respeito do estado do objeto gerenciado). A Figura 88 mostra um exemplo desta situação.

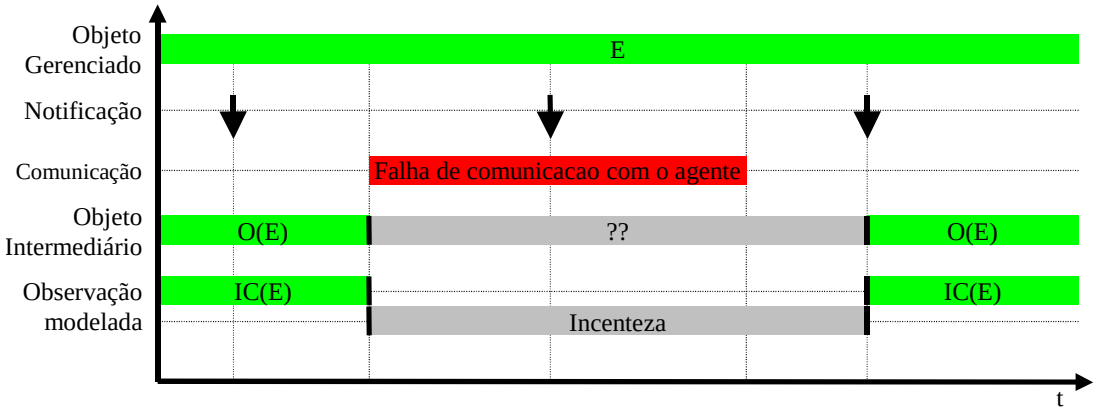


Figura 88 – Exemplo de intervalo de incerteza em uma observação não defasada.

6.5.1.2 Perda de amostragens

A perda de amostragens ou impossibilidade de amostragem é de conhecimento do sistema de monitoramento. Nos sistemas tradicionais, geralmente esta informação não é utilizada. Contudo, ela pode ser de extrema importância e pode ser utilizada no modelamento da observação.

Modelamento para observação defasada em até 1 ciclo

O intervalo de incerteza associado a uma observação defasada em até 1 ciclo, cujo período de amostragem é P, decorrente de uma perda de amostragem ocorrida no instante T_p é definido pela seguinte expressão:

$$II = < T_p - P , T_p + P >$$

A Figura 89 ilustra a ocorrência desta situação.

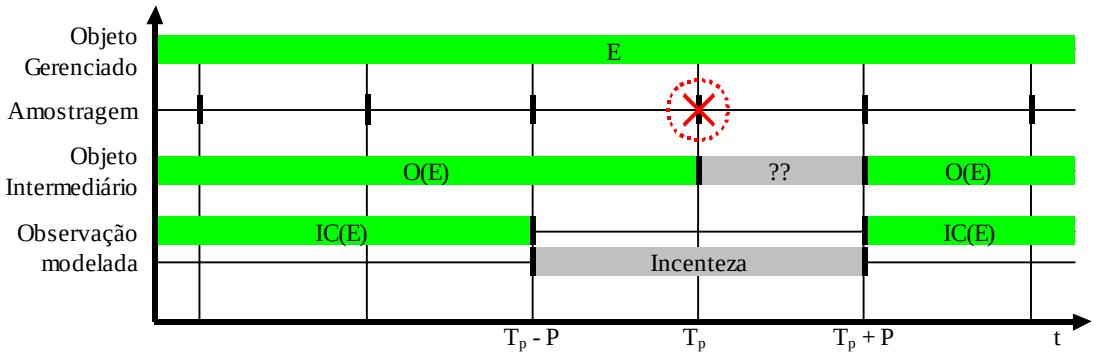


Figura 89 – Exemplo de intervalo de incerteza gerado devido a falta de uma amostragem em observações defasadas em até 1 ciclo.

É possível perceber que a perda de uma amostragem causa um intervalo de tempo de incerteza com duração de 2 períodos.

Modelamento para observação defasada em até 2 ciclos

O intervalo de incerteza associado a uma observação defasada em até 2 ciclos, cujo período de amostragem é P, decorrente de uma perda de amostragem ocorrida no instante T_p , é definido pela seguinte expressão:

$$II = < T_p - 2P , T_p + P >$$

A Figura 90 mostra a ocorrência desta situação.

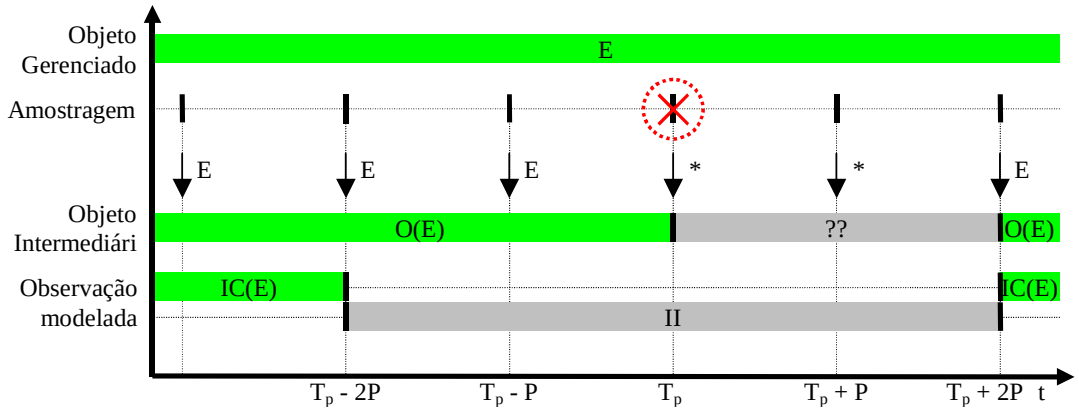


Figura 90 - Exemplo de intervalo de incerteza gerado devido à falta de uma amostragem em observações defasadas em até 2 ciclos.

É importante perceber que no instante T_p + P não é possível definir o estado do objeto gerenciado pois são necessárias duas amostragens (a corrente e a anterior).

É possível perceber que a perda de uma amostragem causa um intervalo de tempo de incerteza com duração de 4 períodos. Fica explícito, portanto, o impacto causado por perda de amostragens neste tipo de situação. Se, por exemplo, um objeto gerenciado possui período de amostragem de 5 minutos, a perda de uma amostragem causa o desconhecimento do estado atual do objeto gerenciado por 20 minutos.

6.5.2 Intervalo de incerteza gerado pela proximidade ao instante corrente

A incerteza do estado atual do objeto gerenciado próximo ao instante corrente ocorre nas observações que se utilizam da técnica de amostragem devido ao desconhecimento do estado do objeto gerenciado entre o instante corrente e a última amostragem. O estado atual será conhecido somente no próximo instante de amostragem.

6.5.2.1 Modelamento para observação não defasada

As observações não defasadas não são afetadas pelo problema de incerteza próximo ao momento corrente.

6.5.2.2 Modelamento para observação defasada em até 1 ciclo

Em observações defasadas em até 1 ciclo, existe um total desconhecimento do estado do objeto gerenciado no intervalo compreendido entre o instante da última amostragem e o instante corrente. A Figura 91 e a Figura 92 exemplificam esta situação mostrando a situação em um instante T_{c1} e a situação em um instante posterior $T_{c2} = T_{c1} + P$, sendo P o período de amostragem.

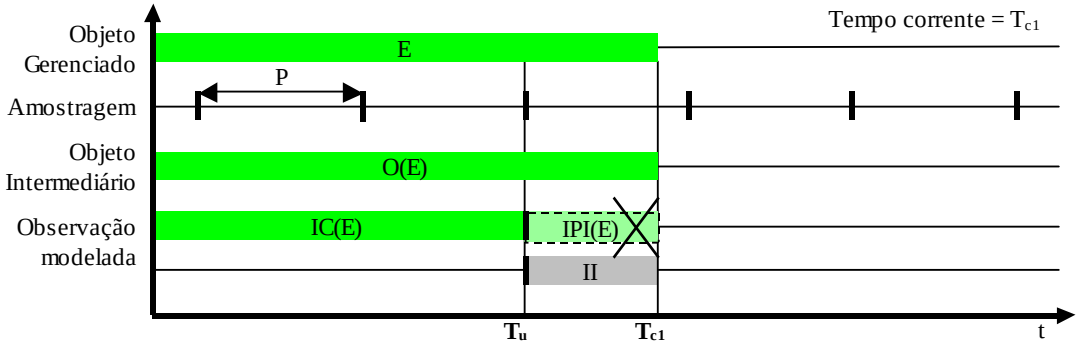


Figura 91 – Exemplo de intervalo de incerteza em uma observação defasada em até 1 ciclo, no instante T_{c1} .

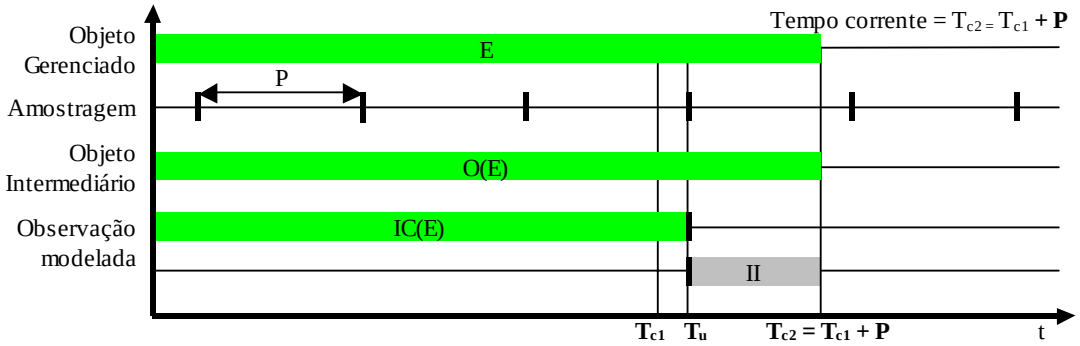


Figura 92 - Exemplo de intervalo de incerteza em uma observação defasada em até 1 ciclo, no instante $T_{c2} = T_{c1} + P$.

A intuição levaria a inserir um intervalo de possibilidade de término de estado "E" entre o instante da última amostragem e o instante corrente. Porém, o intervalo de término de estado indicaria que necessariamente ocorreria uma transição de estado neste intervalo, o que não é necessariamente verdade. Por este motivo deve constar somente o intervalo de incerteza neste intervalo de tempo.

O intervalo de incerteza (II) para uma observação defasada em até 1 ciclo, cujo último instante de amostragem foi T_u , sendo T_c o instante corrente, pode ser definido pela seguinte expressão:

$$II = < T_u, T_c >$$

6.5.2.3 Modelamento para observação defasada em até 2 ciclos

Em observações defasadas em até 2 ciclos, existe um total desconhecimento do intervalo compreendido entre o instante da penúltima amostragem e o instante

corrente. A Figura 93 e Figura 94 exemplificam esta situação mostrando a situação em um instante T_{c1} e a situação em um instante posterior $T_{c2} = T_{c1} + P$, sendo P o período de amostragem.

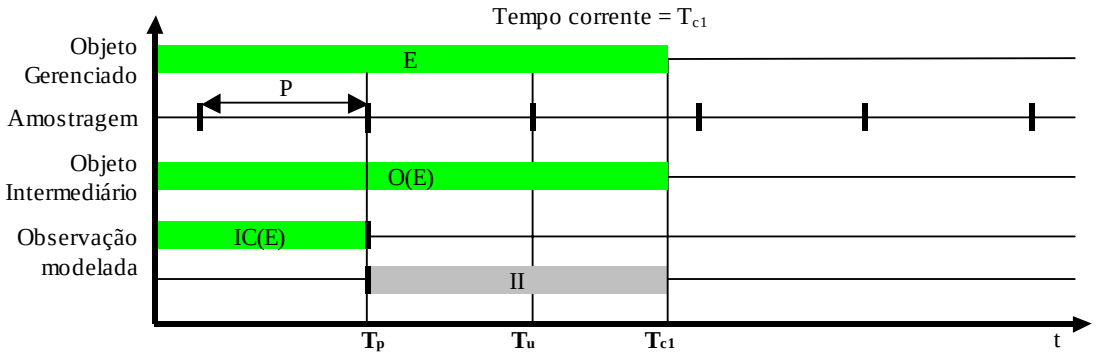


Figura 93 - Exemplo de intervalo de incerteza em uma observação defasada em até 2 ciclos, no instante T_{c1} .

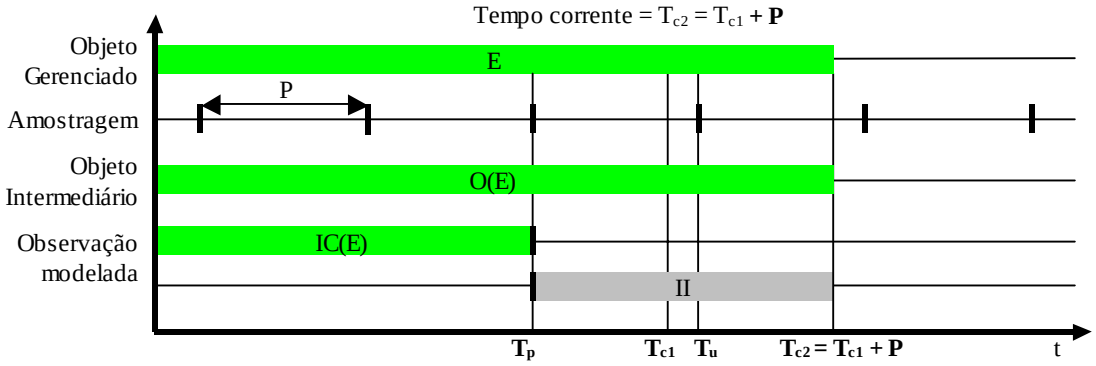


Figura 94 - Exemplo de intervalo de incerteza em uma observação defasada em até 2 ciclos, no instante $T_{c2} = T_{c1} + P$.

O intervalo de incerteza (II) para uma observação defasada em até 2 ciclos cujo penúltimo instante de amostragem foi T_p , sendo T_c o instante corrente, pode ser definido pela seguinte expressão:

$$II = < T_p, T_c > , \text{ sendo } T_p = T_c - P$$

Observação: Em determinados sistemas de monitoração cujo período de amostragem não seja constante (ou seja, apresente variações ou seja ajustável (MOGHÉ, 1998), pode ser conveniente manter o instante de ocorrência das duas últimas amostragens ao invés de manter o valor do período de amostragem P.

6.5.3 Intervalo de incerteza existente no início do processo de monitoração

O intervalo de incerteza está também presente no momento de início do processo de monitoração. O cálculo deste intervalo difere das observações defasadas em até 1 ciclo das defasadas em até ciclos.

6.5.3.1 Nas observações defasadas em até 1 ciclo

Nas observações defasadas em até 1 ciclo, no qual o instante inicial do processo de monitoração é T_i e o instante da ocorrência da primeira amostragem é T_{A1} , o intervalo de incerteza é dado por:

$$II = < T_i , T_{A1} >$$

A Figura 95 mostra um exemplo da ocorrência deste intervalo de incerteza.

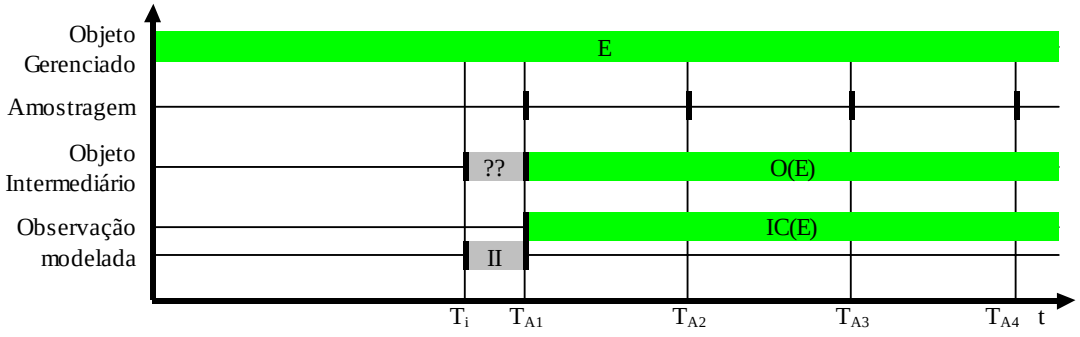


Figura 95 – Exemplo de intervalo de incerteza existente no início do processo de monitoração para observações defasadas em até 1 ciclo.

6.5.3.2 Nas observações defasadas em até 2 ciclos.

Nas observações defasadas em até 2 ciclos, no qual o instante inicial do processo de monitoração é T_i e o instante da ocorrência da segunda amostragem é T_{A2} , o intervalo de incerteza é dado por:

$$II = < T_i , T_{A2} >$$

A Figura 96 mostra um exemplo da ocorrência deste intervalo de incerteza

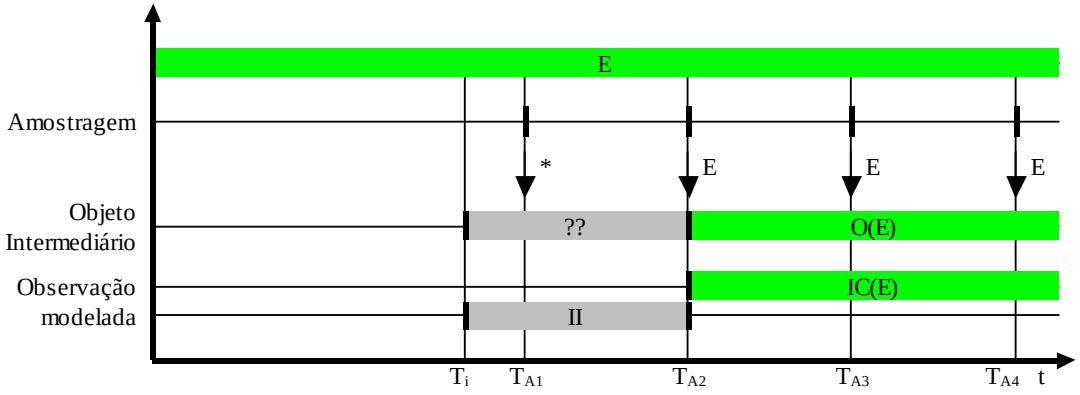


Figura 96 - Exemplo de intervalo de incerteza existente no início do processo de monitoração para observações defasadas em até 1 ciclo.

É importante perceber que no instante T_{A1} não é possível definir o estado do objeto gerenciado, pois são necessárias duas amostragens (a corrente e a anterior que inexiste).

6.6 Modelamento da observação: atrasos de comunicação

É possível também modelar os possíveis atrasos que podem ocorrer nos eventos assíncronos e nas amostragens realizadas. É possível que tais observações, por problemas de carga de processamento, latência de comunicação ou outro motivo qualquer, cheguem atrasadas ao sistema de diagnóstico.

Este atraso pode ser modelado se for possível definir um valor máximo para o atraso da observação de um objeto gerenciado. A Figura 97, Figura 98 e Figura 99 mostra graficamente como este modelamento para observações cuja referência de tempo é local à entidade gerenciadora do objeto gerenciado.

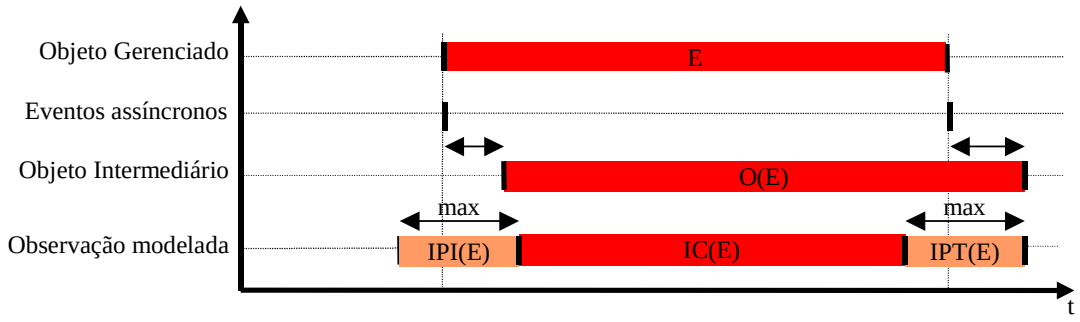


Figura 97 – Exemplo de modelamento de atraso para observações não defasadas.

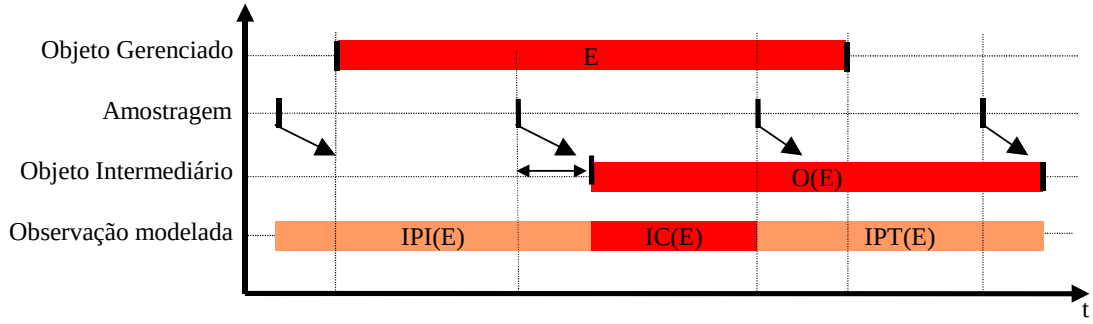


Figura 98 – Exemplo de modelamento de atraso para observações defasadas em até 1 ciclo.

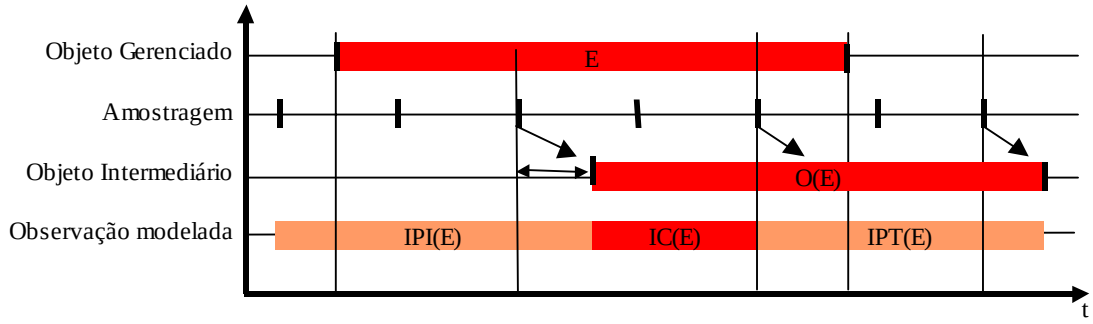


Figura 99 - Exemplo de modelamento de atraso para observações defasadas em até 2 ciclos.

Apesar do atraso poder afetar o intervalo de tempo da observação, ele possui um impacto pequeno porque pode ser considerado desprezível na maior parte dos casos.

6.7 Algoritmo para modelamento da observação

Foi apresentado anteriormente uma técnica de modelamento da observação utilizando os intervalos de certeza, possibilidade e incerteza. A seguir, estão descritos

algoritmos que permitem gerar tais observações para um sistema de diagnóstico. Estes algoritmos são implementados pelo Modelador. A Figura 100 mostra a localização do Modelador no sistema.

O Modelador é um módulo de *software* responsável por realizar o modelamento das observações. O gerente recebe as informações resultantes do processo de monitoração (eventos assíncronos e amostragens periódicas) e atualiza o estado dos objetos intermediários. O Modelador utiliza esses valores de estados como entrada e gera como resultado os intervalos de observação que são armazenados em um banco de dados que fica disponível para o sistema de diagnóstico.

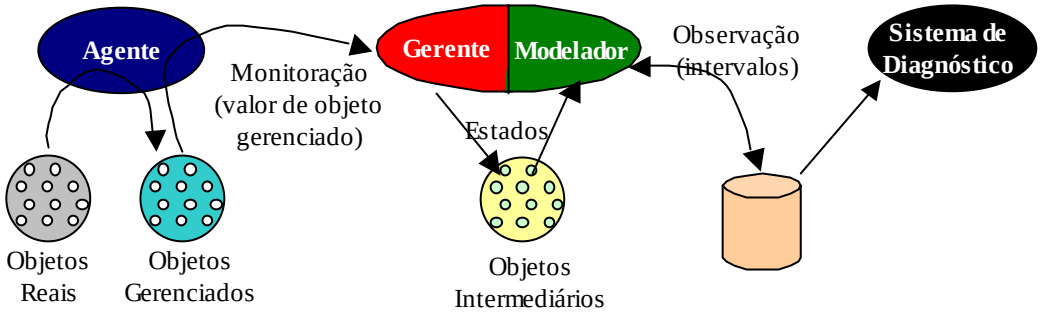


Figura 100 – Posicionamento do Modelador na arquitetura do sistema.

O Modelador, a cada nova informação de estado do objeto intermediário, refina o resultado do modelamento.

```
TC  : Constante indicativa do “Tempo Corrente”
IC  : Constante “Intervalo de Certeza”
IPI : Constante “Intervalo de possibilidade de início”
IPT : Constante “Intervalo de possibilidade de término”
ec  : variável estado corrente
ea  : variável estado anterior
ti  : variável tempo inicial
tf  : variável tempo final
i   : variável número do intervalo
oi  : variável objeto gerenciado
```

Figura 101 – Principais constantes e variáveis utilizadas nos algoritmos.

A Figura 101 mostra algumas constantes e variáveis são utilizadas nos algoritmos apresentados a seguir.

6.7.1 Modelamento de observação não defasada

O modelamento de observações não defasadas é o mais simples de ser modelado. O algoritmo utilizado está descrito na Figura 102.

```
ModelarObservaçãoNãoDefasada(oi)
    {
    i = 1;      // número do intervalo
    (ec,t) = ObterEstado(oi);
    InserirIntervalo(oi,i,IC,e,t,TC);
    i = i + 1;
    repetir
        {
        ea = ec;
        (ec,t) = AguardarAlarme(oi);
        se (perda de comunicação) então
            {
            InserirIntervalo(oi,i,II,NULL,t,TC);
            i = i + 1;
            }
        senão se (ea != ec) então
            {
            (tipo,ea,ti,tf) = RetirarIntervalo(oi,i-1);
            InserirIntervalo(oi,i-1,tipo,ea,ti,t);
            InserirIntervalo(oi,i,IC,ec,t,TC);
            i = i + 1;
            }
        }
    }
```

Figura 102 – Algoritmo de modelamento de observações imediatas.

6.7.2 Modelamento para observações defasadas em até 1 ciclo

A Tabela 3 mostra as ações de modelamento realizadas dependendo da seqüência de estados recebidas pelo Modelador sobre o objeto intermediário. Nesta tabela, “*” significa ausência de conhecimento do estado, “?” significa qualquer estado (inclusive “*”) e “E” representa um estado qualquer distinto do estado “F”.

Seqüência de estados obtidos na amostragem (anterior + corrente)	**Ações para geração dos intervalos.**
* E	II anterior é restrito em t IC(E) pontual é inserido, II posterior é inserido
E E	II anterior é retirado IC(E) anterior é estendido II posterior é inserido
F E	II anterior é retirado IPT(F) anterior é inserido IPI(E) anterior é inserido IC(E) pontual é inserido II posterior é inserido
? *	II anterior é estendido

Tabela 3 – Ações a serem tomadas de acordo com a seqüência de estados obtidos na amostragem.

O algoritmo para modelamento de observações defasadas em até 1 ciclo é mostrado na Figura 103.

```
ModelarObservaçãoDefasadaEmAté1Ciclo(oi)
    {
    i = 1;      // número do intervalo
    t = TempoCorrente();
    ea = NULL;
    ec = NULL;
    InserirIntervalo(oi,i,II,NULL,t,TC);
    i = i + 1;
    repetir
        {
        ea  = ec;
        (ec,t) = AguardarObservaçãoIntermediária(oi);
        se (ec == NULL) então  // (?*)
            {
            // Estender o intervalo de incerteza
            // Nada a ser feito
```

```
            }
         senão se (ea == NULL) então // (* E)
            {
            // II anterior é restrito em t,
            // IC(E) pontual é inserido,
            // II posterior é inserido
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
            InserirIntervalo(oi,i-1,tipo,e,ti,t);
            InserirIntervalo(oi,i,IC,ec,t,t);
            InserirIntervalo(oi,i+1,II,NULL,t,TC);
            i = i + 2;
            }
         senão se (ea == ec) então // (EE)
            {
            // II anterior é retirado, IC(E) anterior
            //  é estendido, II posterior é inserido
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-2);
            InserirIntervalo(oi,i-2,tipo,e,ti,t);
            InserirIntervalo(oi,i-1,II,NULL,t,TC);
            }
        senão // (FE)
            {
            // II anterior é retirado, IPT(F) anterior é
            //  inserido, IPI(E) anterior é inserido, IC(E)
            //  pontual é inserido, II posterior é inserido
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
            InserirIntervalo(oi,i-1,IPT,ea,ti,t);
            InserirIntervalo(oi,i  ,IPI,ec,ti,t);
            InserirIntervalo(oi,i+1,IC,ec,t,t);
            InserirIntervalo(oi,i+2,II,NULL,t,TC);
            i = i + 3;
            }
         }
    }
```

Figura 103 – Algoritmo de modelamento para observações defasadas em até 1 ciclo.

6.7.3 Modelamento para observações defasadas em até 2 ciclos

O algoritmo para modelamento de observações defasadas em até 1 ciclo está mostrado na Figura 104.

A Tabela 4 mostra as ações de modelamento realizadas dependendo da seqüência de estados recebidas pelo Modelador sobre o objeto intermediário. Nesta tabela, '*"significa ausência de conhecimento do estado, "?" significa qualquer estado (inclusive "*") e "E" representa um estado qualquer distinto do estado "F".

Seqüência de estados obtidos na amostragem (pré-anterior, anterior e corrente)	**Ações**
E E E	IC(E) é estendido II iniciado a –2P é retirado II é inserido (de -P a TC)
F E E	II iniciado a –2P é retirado IC(E) pontual é inserido em –P II é inserido (de –P a TC)
* E E	II anterior é restrito em –P IC(E) pontual é inserido em –P II é inserido (de –P a TC)
E F E F F E	II iniciado a –2P é retirado IPT(F) é inserido entre –2P e t IPI(E) é inserido entre –2P e t II é inserido (de –P a TC)
* F E	II anterior é restrito em –P IPT(F) é inserido entre –2P e t IPI(E) é inserido entre –2P e t II é inserido (de –P a TC)
? ? * ? * E	II anterior é estendido

Tabela 4 - Ações a serem tomadas de acordo com a seqüência de estados obtidos na amostragem.

```
ModelarObservaçãoDefasadaEmAté2Ciclos(og)
    {
    i = 1;      // número do intervalo
    tc = TempoCorrente();
    ta = tc;
    taa = ta;
    ec = NULL;
    ea = NULL;
    eaa = NULL;
    InserirIntervalo(oi,i,II,NULL,tc,TC);
    i = i + 1;
    repetir
        {
        eaa = ea;
        ea  = ec;
        ta  = tc
        taa = ta;
        (ec,tc) = AguardarObservaçãoIntermediária(oi);
        se (ec == NULL) então  // (? ?*)
            {
            // Estender o intervalo de incerteza
            // nada a ser feito
            }
        senão se (ea == NULL) então  // (? * E)
            {
            // Estender o intervalo de incerteza
            // nada a ser feito
            }
        senão se (eaa == NULL) & (ea == ec) então  // (* E E)
            {
            // II anterior é restrito em –P
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
            InserirIntervalo(oi,i-1,tipo,e,ti,ta);
            // IC(E) pontual é inserido em –P
            InserirIntervalo(oi,i,IC,ec,ta,ta);
            // II é inserido (de –P a TC)
            InserirIntervalo(oi,i+1,II,NULL,ta,TC);
            i = i + 2;
```

```
        }
    senão se (eaa == NULL) & (ea != ec) então  // (* F E)
        {
        // II anterior é restrito em tc–P
        (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
        InserirIntervalo(oi,i-1,tipo,e,ti,ta);
        // IPT(F) é inserido entre tc–2P e tc
        InserirIntervalo(oi,i,IPT,ea,taa,tc);
        // IPI(E) é inserido entre tc–2P e tc
        InserirIntervalo(oi,i+1,IPI,ec,taa,tc);
        // II é inserido (de tc–P a TC)
        InserirIntervalo(oi,i+2,II,NULL,ta,TC);
        i = i + 3;
        }
    senão se  (ea != ec) então  // (E F E, F F E)
        {
        // II iniciado a –2P é retirado
        (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
        // IPT(F) é inserido entre tc–2P e tc
        InserirIntervalo(oi,i-1,IPT,ea,taa,tc);
        // IPI(E) é inserido entre tc–2P e tc
        InserirIntervalo(oi,i  ,IPI,ec,taa,tc);
        // II é inserido (de tc–P a TC)
        InserirIntervalo(oi,i+1,II,NULL,ta,TC);
        i = i + 3;
        }
    senão se  (ea == ec) & (eaa == ea) então  // (E E E)
        {
        // IC(E) é estendido
        (tipo,e,ti,tf) = RetirarIntervalo(oi,i-2);
        InserirIntervalo(oi,i-2,tipo,e,ti,ta);
        // II iniciado a tc–2P é retirado
        (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
        // II é inserido (de tc-P a TC)
        InserirIntervalo(oi,i-1,IPT,ea,ta,TC);
        }
    senão se  (ea == ec) & (eaa != ea) então  // (F E E)
        // II iniciado a –2P é retirado
```

```
            (tipo,e,ti,tf) = RetirarIntervalo(oi,i-1);
            // IC(E) pontual é inserido em –P
            InserirIntervalo(oi,i-1,IC,ec,ta,ta);
            // II é inserido (de tc-P a TC)
            InserirIntervalo(oi,i,II,NULL,ta,TC);
        senão
            Erro
        }
    }
```

Figura 104 – Algoritmo de modelamento para observações defasadas em até 2 ciclos.

Obs: É suposto que a primeira amostragem (que não retorna valor pois depende de uma amostragem anterior inexistente) retorne NULL.

6.8 Aglomerado (*cluster*) de intervalos

Uma atividade realizada freqüentemente em um sistema de diagnóstico é a verificação se um conjunto de sintomas está associado a uma determinada anomalia. Em um sistema atemporal, basta verificar se existe algum relacionamento causal entre os sintomas e a anomalia. Em um sistema temporal, além de verificar o relacionamento causal é necessário verificar também se a localização da ocorrências dos sintomas no tempo são consistentes.

No modelamento temporal da observação proposto neste trabalho, a ocorrência da observação de um determinado estado é representada por uma seqüência de intervalos. A esta seqüência de intervalos será dado o nome de aglomerado.

Definição 19: Aglomerado (*cluster*) de intervalos de observação

Um aglomerado (*cluster*) de intervalos de observação associado à uma ocorrência E_i de um estado E em um objeto gerenciado é definido como sendo a seqüência de intervalos de observação representativo para caracterizar o intervalo da ocorrência E_i.

A Figura 105 mostra um exemplo no qual são mostrados os *clusters* de observação de um objeto gerenciado cuja observação é defasada em até 1 ciclo e a Figura 106, defasada em até 2 ciclos.

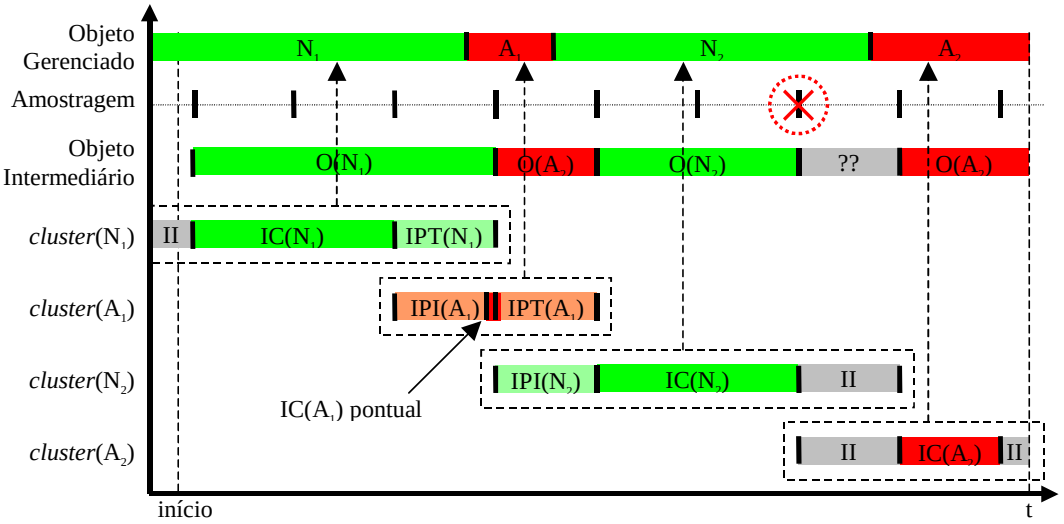


Figura 105 – Exemplos de *clusters* em observação defasada em até 1 ciclo.

A construção de um *cluster* é realizada a partir da ordem de criação dos intervalos, sendo o *cluster* o agrupamento de uma seqüência de intervalos IPI, IC, IPT ou II todos associados à mesma ocorrência Ei de um estado no objeto gerenciado.

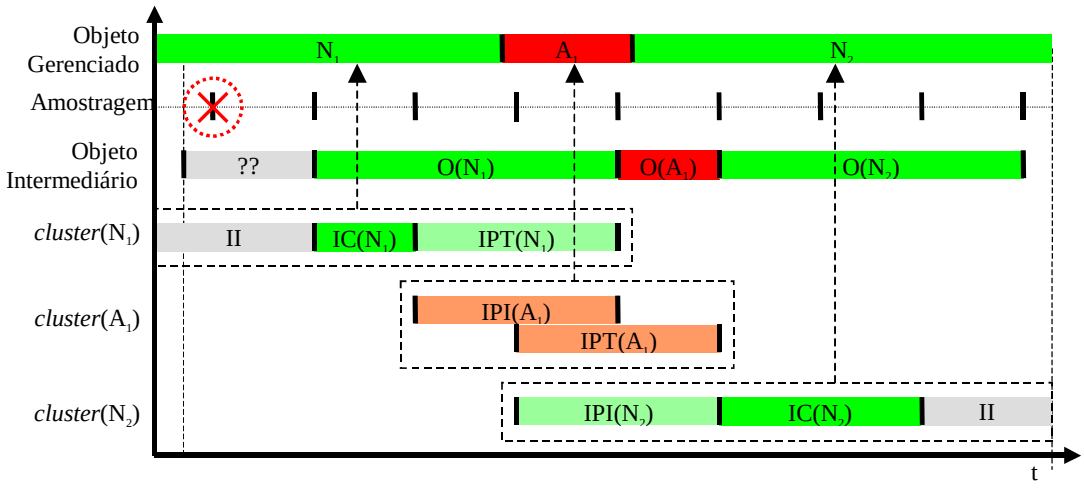


Figura 106 – Exemplos de *clusters* em observação defasada em até 2 ciclos.

Existe uma situação particular na qual um intervalo de incerteza (II) pode ser compartilhado por dois *clusters*, como mostrado na Figura 105.

6.9 Forma normal para *cluster*

O processo de raciocínio de um sistema de diagnóstico opera diretamente sobre *clusters*, pois estes estão diretamente associados à ocorrência de um determinado estado em um objeto gerenciado. Desta forma é necessário definir alguns operadores para manipulação de clusters.

Existem diversas seqüências de intervalos que podem ser representações de clusters. Porém, por simplificação, alguns operadores podem ser capazes de operar somente sobre um subconjunto das forma de cluster possível.

A forma mais geral para representação de cluster é chamada aqui de "forma normal" e será utilizada na definição de alguns operadores.

Definição 20: Cluster formato normal

Um cluster está no formato normal se estiver em uma das seguintes formas:

C= (IPI, IC, IPT)

Assim, na Figura 106 pode ser considerado um *cluster* formato normal somente o *cluster*(A_1).

6.9.1 O processo de normalização

O processo de normalização tem por objetivo simplificar e uniformizar o formato do *cluster* para a forma normal. Existem diversas formas de realizar um processo de normalização, cada uma com um processo diferente de aproximação. Geralmente, elas levam em conta principalmente como serão aproximados os intervalos de incerteza.

A Figura 107 mostra uma das possíveis formas de realizar a normalização.

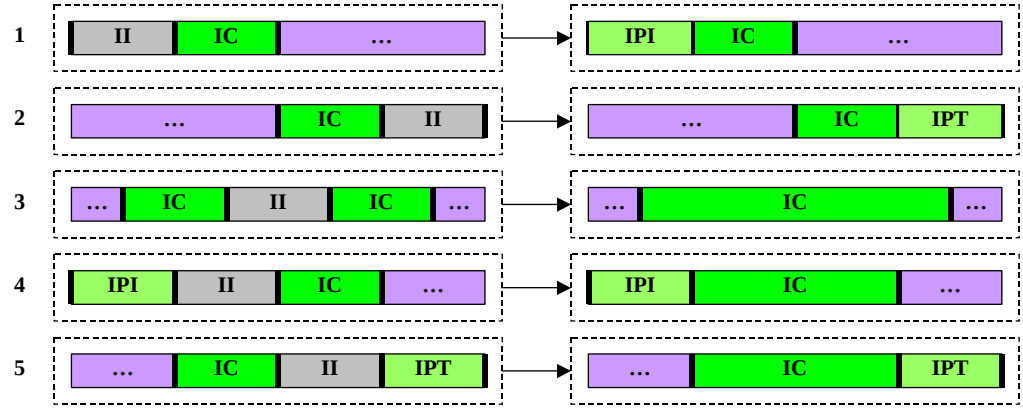


Figura 107 – Uma das possíveis formas de realizar a normalização.

A normalização, apesar de ser uma aproximação que pode acrescentar erros, é importante para facilitar a manipulação dos *clusters*.

A normalização é um processo opcional. Foi incluído com o objetivo de facilitar o entendimento. Caso contrário, seria necessário incluir diversas outras alternativas de configuração para um *cluster* nas definições a seguir.

6.10 Intersecção temporal entre *clusters* na forma normal

Definição 21: Intersecção entre dois *clusters* na forma normal

A Intersecção entre dois *clusters* C1 e C2 na forma normal, denotada por C1 ∩ C2, é definida como sendo uma operação que resulta em um *cluster* C que representa todos os intervalos possíveis que ainda satisfaçam a C1 e C2. Caso não exista tal intervalo C é dito inconsistente e resulta em uma seqüência vazia.

Sejam dois *clusters* na forma normal C1 = (IPI(S1), IC(S1) e IPT(S1)) e C2 = (IPI(S2), IC(S2) e IPT(S2)). O *cluster* C = (IPI(C), IC(C), IPT(C)), resultado da intersecção de C1 e C2, se inconsistente:

- IPI(S1) ∩ IPI(S2) = { } ou
- IPT(S1) ∩ IPT(S2) = { },

Se consistente (IPI(S1) ∩ IPI(S2) ≠ { } e IPT(S1) ∩ IPT(S2) ≠ { }) é também um *cluster* na forma normal e pode ser representado por:

- IPI(C) = IPI(S1) ∩ IPI(S2);
- IC(C) = IC(S1) ∪ IC(S2);
- IPT(C) = IPT(S1) ∩ IPI(S2).

6.11 Intersecção temporal em relações causais entre anomalia e sintoma

Até o presente momento, a teoria de intervalos de observação sempre esteve associada às observações. Chegou o momento de utilizar esta teoria para representar os estados dos componentes do sistema. A Figura 108 ilustra uma relação causal entre anomalia e sintoma.

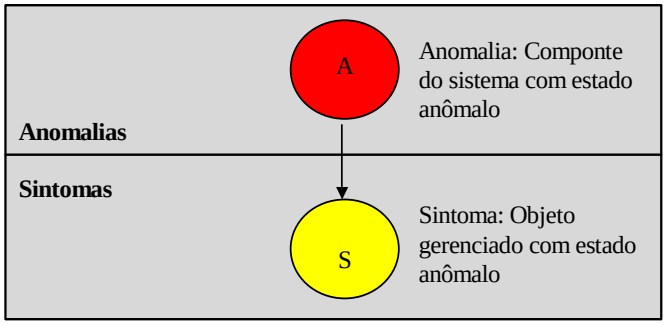


Figura 108 – Relação causal entre uma anomalia e um sintoma.

Supondo que esta relação seja do tipo “imediata e necessariamente causa” e conhecendo o intervalo de tempo aproximado da ocorrência S é possível supor o intervalo de tempo aproximado da anomalia A.

Seja a relação “A imediatamente e necessariamente causa S”. Seja também Sj uma ocorrência do sintoma S e C(Sj) o cluster associado a esta ocorrência. Se o sintoma Sj for causado pela anomalia Ai , então

- C(Ai) = C(Sj) , é um cluster válido para representar a ocorrência de Ai.

Também é importante a situação na qual uma anomalia pode causar mais que um sintoma, mostrada na Figura 109. Neste caso, através dos intervalos de ocorrência

dos sintomas é possível inferir o intervalo de ocorrência da anomalia, caso esta seja a causadora dos sintomas.

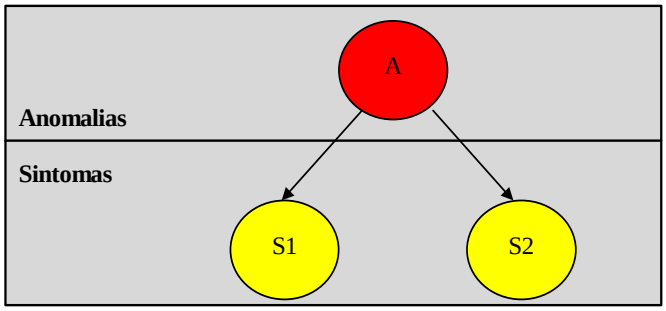


Figura 109 - Relação causal entre uma anomalia e dois sintoma.

Seja a relação “A imediatamente e necessariamente causa S_1” e a relação “A imediatamente e necessariamente causa S_2”. Sejam S_{1j} e S_{2k} ocorrências dos sintomas S_1 e S_2 respectivamente e também $C(S_{1j})$ = (IPI(S_{1j}), IC(S_{1j}) e IPT(S_{1j})) e $C(S_{2k})$ = (IPI(S_{2k}), IC(S_{2k}) e IPT(S_{2k})) os clusters associados a estas ocorrências. Se a anomalia A_i for a causadora de S_{1j} e S_{2k} então:

- $C(S_{1j}) \cap C(S_{2k})$ é consistente e;
- $C(A_i) = C(S_{1j}) \cap C(S_{2k})$

Caso a interseção não seja consistente significa que não atende à relação causal (causa imediata e necessariamente) não sendo Ai a causa de S_1 e S_2. Os motivos para que os sintomas não estejam associados a anomalia A podem ser:

- S_{1j} e S_{2k} não foram causados por Ai;
- Presença de ruído em uma das observações (sintomas).

6.12 Intersecção temporal em relações causais entre anomalias

A mesma teoria definida para o relacionamento entre anomalia e sintoma vale também para o relacionamento entre anomalias, como mostrado na Figura 110 e Figura 111.

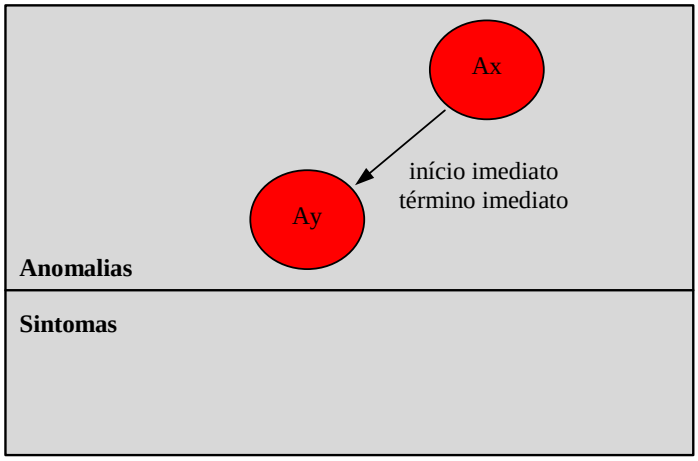


Figura 110 – Relação causal entre duas anomalias.

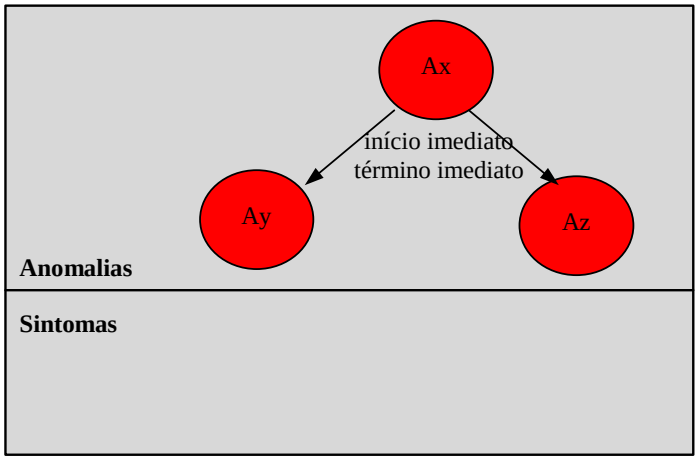


Figura 111 – Relações causais entre anomalias

6.13 Conclusão

Este capítulo descreveu uma possível forma de modelamento da observação, de forma a incorporar informações a respeito das imprecisões temporais e incertezas de observação. Também foram apresentados algoritmos que possibilitam modelar tais observações, mostrando a viabilidade prática de implementação.

O modelamento da observação é pré-requisito básico para um sistema de diagnóstico temporal de forma a contornar as imprecisões temporais das observações. Ele será utilizado no sistema de diagnóstico proposto neste trabalho.

7. Modelos e Métodos para diagnóstico temporal

Este capítulo tem como objetivo apresentar um método de diagnóstico temporal que utilize o modelamento de observação proposto.

Para possibilitar avaliar a utilidade do modelamento das observações foi concebido um sistema de diagnóstico temporal baseado em modelo. O diagnóstico é realizado sobre um determinado instante escolhido que leva em consideração a localização no tempo das observações. O instante escolhido pode ser próximo ao instante corrente ou não (caracterizando DMC ou DMP). Caso seja próximo ao instante corrente pode ser ativado um método que possibilita a obtenção de observações adicionais. O diagnóstico considera também a possibilidade de ausência de observação.

O método de diagnóstico utiliza um grafo causal e um conjunto de observações, como mostrado na Figura 112. O grafo causal pode ser obtido, geralmente sem dificuldade, a partir de modelos estruturais, comportamentais e causais.

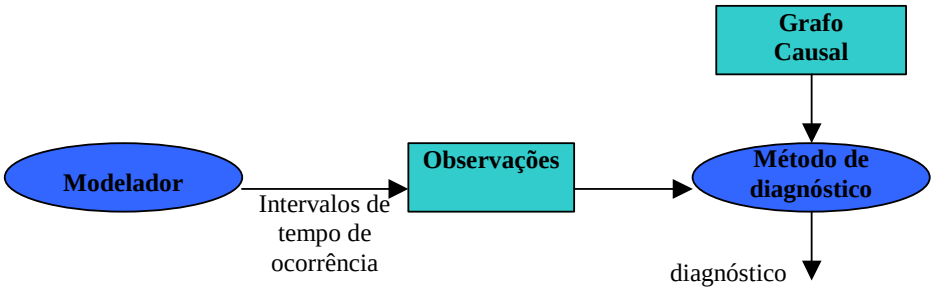


Figura 112 – Interface do método de diagnóstico

7.1 SiDiR-t

O sistema SiDiR-t é um sistema de diagnóstico baseado no sistema SiDiR com a adição de características temporais.

O sistema SiDIR é um protótipo de um sistema de diagnóstico desenvolvido na Universidade de São Paulo (BARROS, 1999; BERNAL, 1999b; LEMOS 1999;

FONTANINI 2002) que será utilizado como referência para mostrar a viabilidade do método de diagnóstico proposto. É um sistema baseado em modelos interpretativos que permitem descrever parte do comportamento e topologia do ambiente computacional.

7.2 Diagnóstico temporal utilizado no SiDiR-t

A classe de diagnóstico utilizada no SiDiR-t é uma variante do diagnóstico temporal. A partir de um instante de diagnóstico escolhido são selecionados os intervalos de observações (*clusters*) que representam sintomas.

É considerado um diagnóstico temporal porque a localização do sintoma no tempo é preservada e utilizada pelo método de diagnóstico.

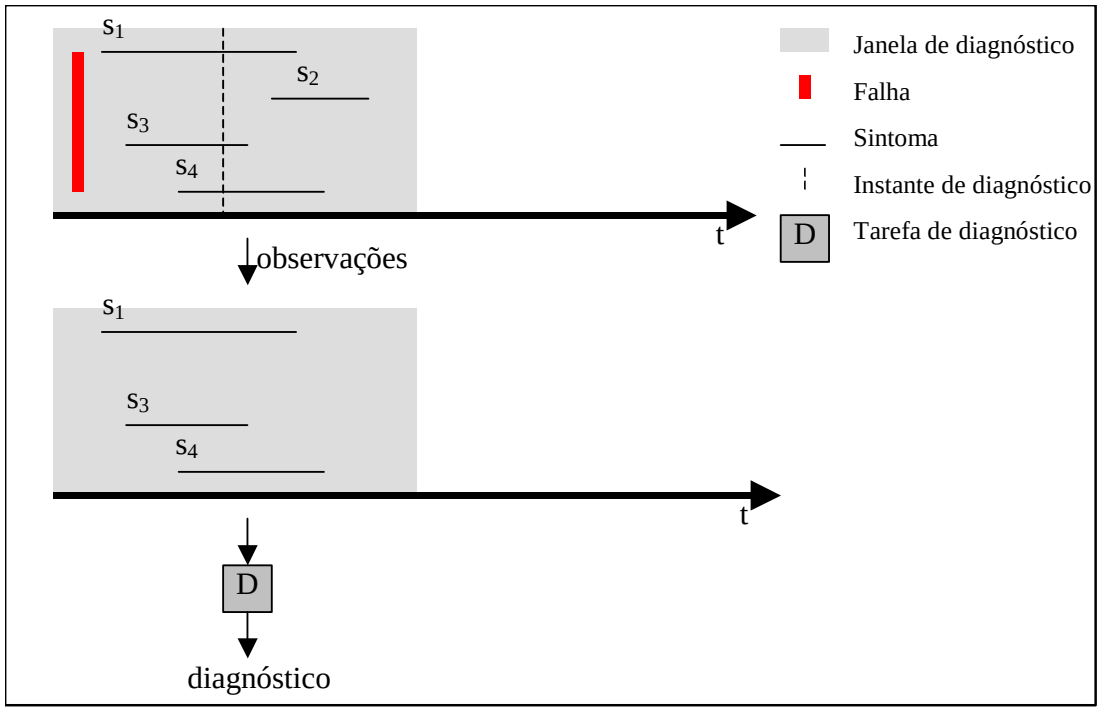


Figura 113 – Modelo de diagnóstico temporal utilizado no sistema SiDiR-t

7.3 Modelos utilizados

O diagnóstico é apoiado diretamente sobre um grafo causal e em um conjunto de observações. Diferentemente do sistema SMARTS, não é gerado um grafo de

correlação bipartido. O sistema de correlação livro-código utilizado pelo sistema SMARTS elimina informações valiosas a respeito do relacionamento entre as anomalias (e entre anomalias e sintomas) quando ele é convertido para um grafo bipartido.

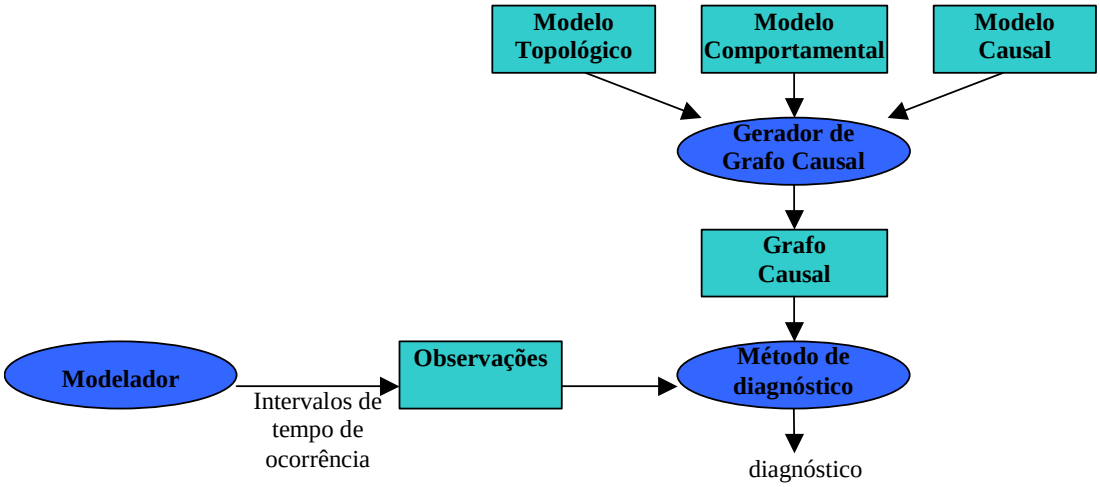


Figura 114 – Modelos utilizados no diagnóstico

O grafo causal pode ser gerado de antemão, antes do início do processo de diagnóstico, e é válido enquanto não for alterada a topologia do ambiente.

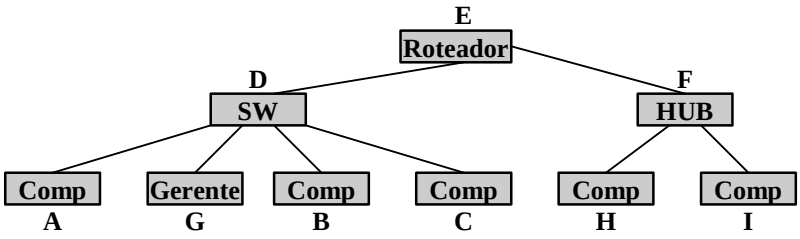


Figura 115 – Exemplo de modelo de configuração do domínio físico

A Figura 115 ilustra um exemplo de modelo de configuração do domínio físico de um ambiente computacional e a Figura 116 o modelo de configuração do domínio de subrede.

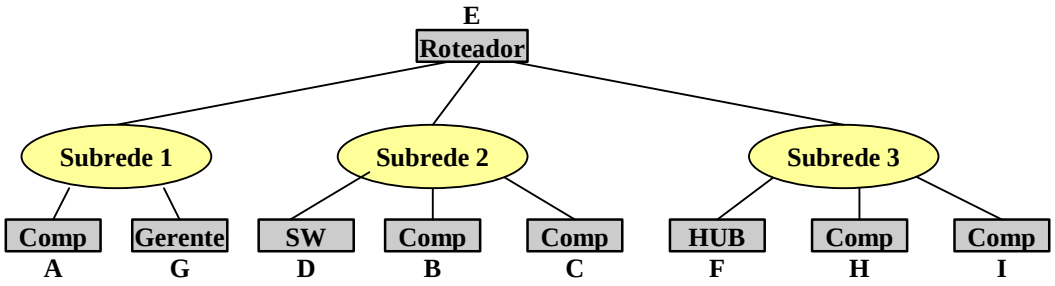


Figura 116 – Exemplo de modelo de configuração de domínio de subrede

O grafo da Figura 117 é resultado da aplicação de um modelo comportamental de comunicação de pacotes IP sobre uma infra-estrutura de comunicação baseada na pilha de protocolos TCP/IP e dos modelos de configuração de domínio físico e de subrede. O grafo explicita a relação causal entre falha de um equipamento (anomalia) e os sintomas (perda de comunicação) do gerente (G) um outro equipamento.

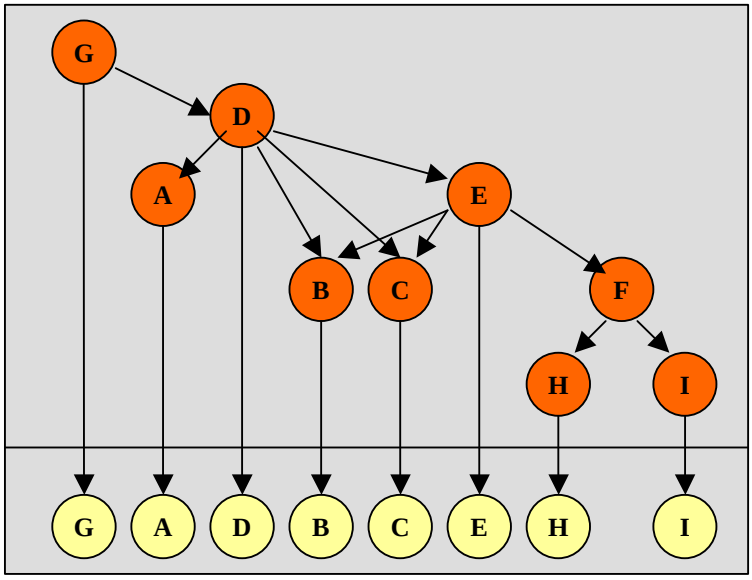


Figura 117 – Exemplo de grafo causal derivado dos modelos

Outro exemplo que pode ser apresentado está mostrado na Figura 118 que mostra a relação causal entre as anomalias que podem ocorrer em um equipamento e os sintomas apresentados.

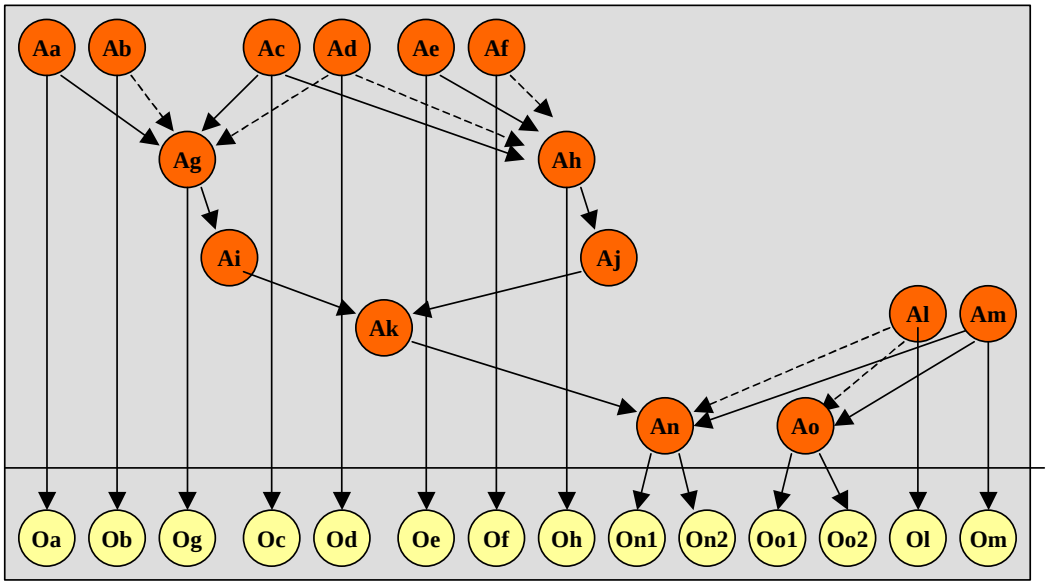


Figura 118 – Exemplo de grafo causal dos sintomas e anomalias de um servidor

A Figura 119 descreve as anomalias e sintomas representadas no grafo. Este grafo pode ser criado a partir das relações causais definidas e do modelo de configuração do sistema (modelo topológico e estrutural).

Tipo	Nó	Descrição
Anomalias	Aa	Ventilador da CPU1 não funciona
	Ab	Velocidade baixa do ventilador da CPU1
	Ac	Ventilador do gabinete não funciona
	Ad	Velocidade baixa do ventilador do gabinete
	Ae	Ventilador da CPU2 não funciona
	Af	Velocidade baixa do ventilador da CPU2
	Ag	Alta temperatura na CPU1
	Ah	Alta temperatura na CPU2
	Ai	Falha na CPU1
	Aj	Falha na CPU2
	Ak	Falha no equipamento
	Al	Sistema de arquivos cheio
	Am	Área de swap cheia
	An	Servidor DNS terminado
	Ao	Servidor WEB terminado
Observações	Oa	Ventilador da CPU1 não funciona
	Ob	Velocidade baixa do ventilador da CPU1
	Oc	Ventilador do gabinete não funciona
	Od	Velocidade baixa do ventilador do gabinete
	Oe	Ventilador da CPU2 não funciona
	Of	Velocidade baixa do ventilador da CPU2
	Og	Alta temperatura na CPU1
	Oh	Alta temperatura na CPU2
	On1	Processo DNS não existe
	On2	Serviço DNS não responde
	Oo1	Processo WEB não existe
	Oo2	Serviço WEB não responde
	Ol	Sistema de arquivos cheio
	Om	Área de swap cheia

Figura 119 – Relação de sintomas e anomalias apresentadas no grafo causal da Figura 118.

O grafo utilizado pode conter também relações do tipo “possivelmente causa”. Também pode conter relações causais retardadas. O método descrito a seguir supõe o uso de um grafo causal do tipo “necessária e imediatamente causa”, exceto quando explicitamente citado o contrário.

7.4 Método de diagnóstico

De maneira geral, o método de diagnóstico verifica inicialmente quais são as possíveis causas para o sintomas apresentados consultando o grafo causal. Em

seguida é verificado se os intervalos de tempo de ocorrência das causas e seus efeitos são consistentes, utilizando o operador intersecção entre *clusters*. As hipóteses inconsistentes são descartadas.

Essa verificação de consistência considera os intervalos de incerteza. Assim, quando não existe observação a respeito de um objeto intermediário não impede que esta ainda possa ser relacionada como hipótese válida (uma possível causa).

7.5 Descrição do método de diagnóstico

Um método de resolução de problema define o modo pelo qual o objetivo de uma tarefa pode ser alcançado através da execução de subtarefas. Benjamins (1993) descreve várias alternativas de modelamento de métodos de resolução de problemas, baseado na visão de que uma tarefa de diagnóstico seja composta por três subtarefas (Figura 120):

- detecção de sintomas;
- geração de hipóteses;
- discriminação de hipóteses.

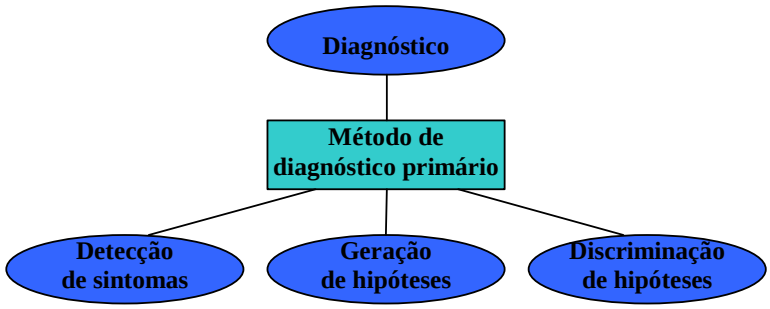


Figura 120 – A tarefa de diagnóstico segundo Benjamins (1993).

Detecção de sintomas: O processo de diagnóstico recebe como entrada um conjunto de observações. Estas observações são analisadas a fim de serem consideradas sintomas ou não.

Geração de hipóteses: Gera um conjunto de hipóteses partindo das observações iniciais, sejam elas normais ou anômalas (sintomas). Pode fazer uso do conhecimento estático (por exemplo um grafo causal) ou dinâmico (uma simulação do

comportamento de um dispositivo em função de determinadas entradas). Não são consideradas aqui tarefas que utilizem de observações adicionais.

Discriminação de hipóteses: O objetivo desta tarefa é reduzir o conjunto de hipóteses pela requisição de informações adicionais a respeito do ambiente diagnosticado para serem relacionados com o conjunto de hipóteses. Hipóteses inconsistentes são excluídas.

7.5.1 Detecção de sintomas

A Figura 121 mostra os métodos propostos por Benjamins (1993) para a tarefa de detecção de sintomas. A linha tracejada indica uma opção, uma alternativa. Neste caso, a tarefa pode utilizar uma das seguintes tarefas: “compara”, “classifica” e “pergunta ao usuário”.

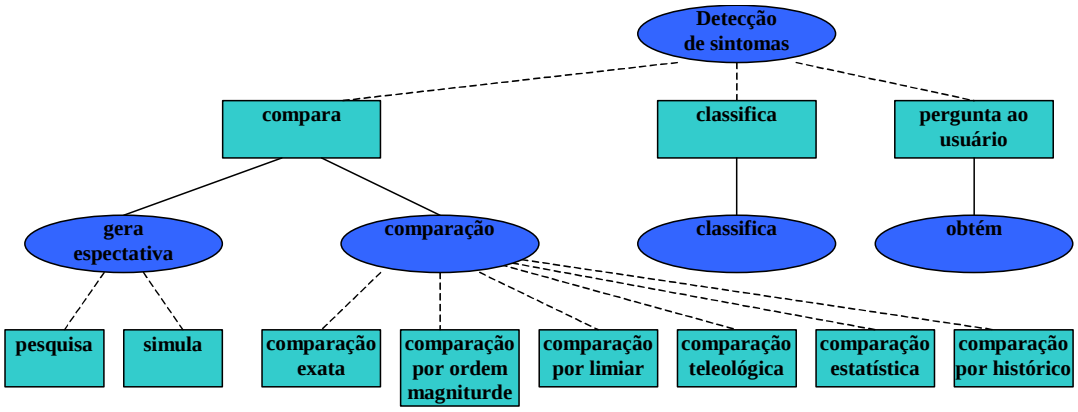


Figura 121 – Métodos propostos por Benjamins (1993) para a tarefa “detecção de sintomas”

No sistema SiDiR-t, parte da tarefa de detecção de sintomas é delegada à entidades de apoio, como o “Mapeador em estados” e o “Modelador”, como mostrado na Figura 122. Estes utilizam diversos métodos descritos em (BENJAMINS, 1993) relacionados à tarefa de detecção de sintomas como:

- comparação exata;
- comparação por ordem de magnitude;
- comparação por limiar.

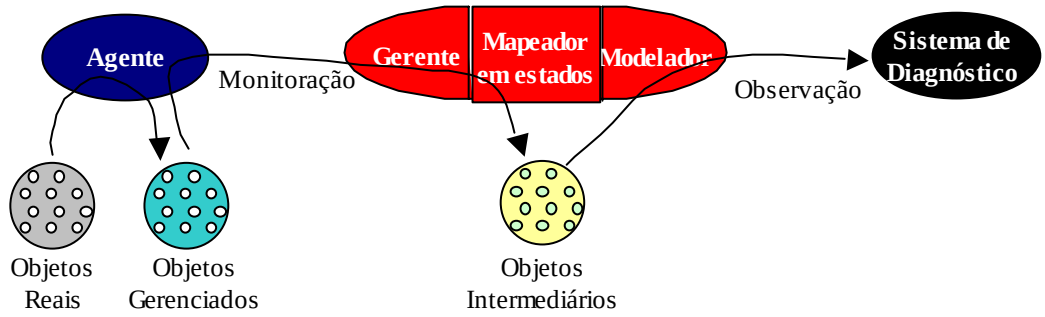


Figura 122 – Papel das entidades de apoio no sistema de diagnóstico.

As observações geradas pelo sistema de apoio são do tipo "intervalo de tempo" [25]. O conjunto de intervalos associados a um objeto intermediário são agrupados formando os aglomerados (*clusters*). Um aglomerado representa um conjunto de intervalos de tempo associados à ocorrência de um determinado estado no objeto intermediário.

Estes aglomerados são formados dinamicamente durante a evolução do sistema. Portanto, é possível considerar que as observações (entradas para o sistema de diagnóstico) são formadas por aglomerados, vários deles incompletos.

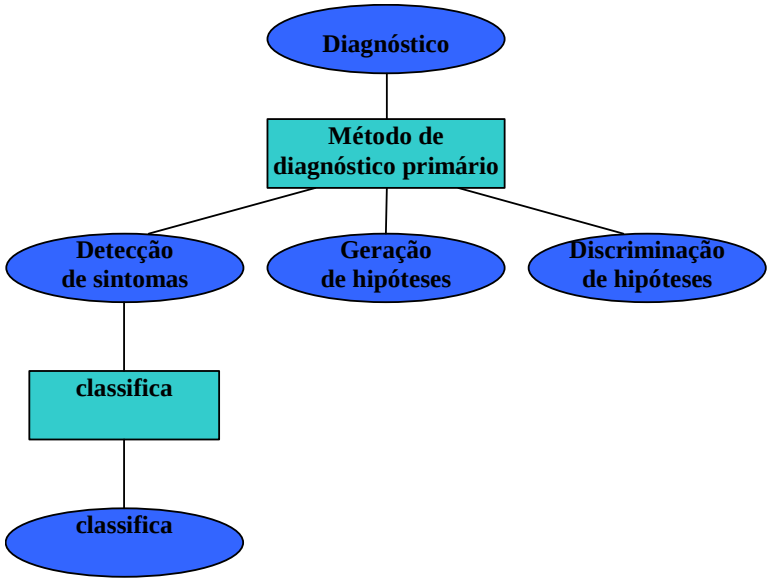


Figura 123 - Métodos utilizado pelo sistema SiDiR-t para a tarefa de detecção de sintomas.

[25] Descrito no capítulo 6.3.1.

No sistema SiDiR-t, a tarefa “classifica” é responsável pela seleção de todos os *clusters* que contenham o instante de diagnóstico. Possui também a função de classificar cada *cluster* como sendo representativo de um estado “anômalo”, estado “normal” ou estado “indeterminado”, gerando três conjuntos de observações: OBS_{normal}, $OBS_{anômala}$.e $OBS_{indeterminada}$ sendo $OBS = OBS_{normal} \cup OBS_{anômala}. \cup OBS_{indeterminada}$. O conjunto $OBS_{anômala}$ também é chamado de conjunto de sintomas $S = OBS_{anômala}$.

Da forma com que as observações são modeladas, pode existir mais que um *cluster* associado a um mesmo objeto intermediário em um mesmo instante de tempo (por exemplo, próximo a uma transição de estados).

A tarefa de detecção de sintomas tem também o papel de ativar a tarefa de geração de hipóteses quando forem detectados sintomas no ambiente.

Para citar um exemplo, considere-se o exemplo da Figura 117. O processo detecção de sintomas pode ter detectado:

- $OBS_{normal} = \{ G, B, C, E\}$;
- $OBS_{anômala}. = \{ A, H, I\}$;
- $OBS_{indeterminada} = \{D\}$.

7.5.2 Geração de hipóteses

A tarefa de geração de hipóteses tem o papel de gerar um conjunto de hipóteses que explique o conjunto de observações, sejam elas normais ou anômalas. Ela pode utilizar um método empírico ou baseado em modelo. O sistema SiDiR-t é baseado em modelo.

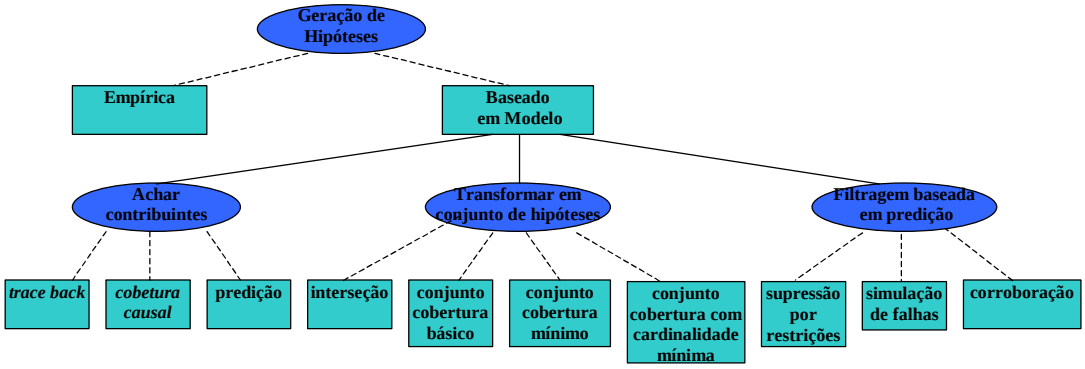


Figura 124 - Métodos propostos por Benjamins (1993) para a tarefa de geração de hipóteses.

A tarefa de geração de hipóteses tem como entrada um conjunto de observações iniciais ($OBS_{inicial} = OBS_{normal} \cup OBS_{anômala}. \cup OBS_{indeterminada}$). Cada observação é representadas na forma de *clusters*. Segundo Benjamins (1993), a tarefa de geração de hipóteses, nos sistemas baseados em modelo, é geralmente subdividida em 3 subtarefas:

- achar contribuintes;
- transformar em conjunto de hipóteses;
- filtragem baseada em predição.

A Figura 125 mostra os métodos de geração de hipóteses utilizados pelo sistema SiDiR-t.

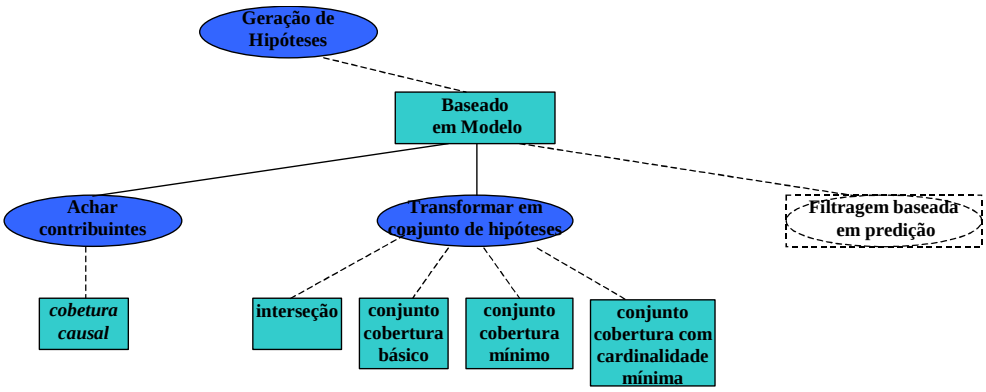


Figura 125 – Métodos de geração de hipóteses utilizado pelo sistema SiDiR-t.

7.5.2.1 Achar contribuintes

O primeiro passo é encontrar um conjunto de contribuintes (tarefa achar contribuintes). Para cada sintoma S_i detectado é calculado o conjunto de anomalias que podem ter ocorrido no ambiente e causado o sintoma S_i.

O sistema utiliza o método de cobertura causal para encontrar os possíveis contribuintes. Para cada sintoma $S_i \in OBS_{anômala}$ é gerado um conjunto contribuinte "c_i" gerando o conjunto $C = \{ c_1, c_2, ..., c_n\}$ dos contribuintes.

O conjunto contribuinte de um sintoma relaciona todas as suas possíveis causas. De todas as relacionadas, provavelmente somente uma é a causa correta.

Como exemplo, é possível citar novamente o exemplo da Figura 117. Para os sintomas detectados é possível construir o conjunto de contribuintes:

- $C = \{ c_a, c_h, c_i \}$;
- $c_a = \{A, D, G\}$;
- $c_h = \{ H, F, E, D, G\}$;
- $c_i = \{ I, F, E, D, G\}$.

7.5.2.2 Transformar em conjunto de hipóteses

O próximo passo é gerar um conjunto de hipóteses inicial (h) a partir dos conjuntos de contribuintes gerados anteriormente. Os principais métodos existentes são:

- **Método de intersecção**: Adequado somente para sistemas que possam apresentar no máximo uma falha por vez;
- **Método de conjunto cobertura**: A solução é qualquer conjunto cuja intersecção com cada conjunto contribuinte não seja vazia. Esta solução não é necessariamente parcimoniosa. O conjunto solução pode ser o próprio conjunto C;
- **Método de subconjunto minimalista**: Semelhante ao anterior porém a solução deve ser minimalista, ou seja, não deve existir um outro subconjunto h' de soluções que também possa ser cobertura dos contribuintes;

- **Método de cardinalidade mínima**: A solução deve ter cardinalidade mínima, mas ainda ser cobertura dos contribuintes.

Em um sistema temporal a informação sobre tempo pode ser utilizada para verificar a consistência temporal. Para isto, dado o subgrafo G' gerado a partir do grafo original G eliminando os vértices não presentes em C, é verificada a consistência temporal do *cluster* de observação (utilizando as operações de intersecção entre *clusters*) associado a cada vértice do subgrafo G. Seja h o conjunto de vértices cujos *clusters* são temporalmente consistentes. Pode ser utilizado, agora, o método de cobertura minimalista para a escolha da solução.

É possível também mensurar a qualidade do diagnóstico (grau de confiança) contabilizando a utilização de intervalos de incerteza no processo de intersecção dos *clusters* de observação.

7.5.2.3 Filtragem baseada em predição

Não são utilizados métodos de filtragem baseada em predição.

7.5.3 Discriminação de hipóteses

A tarefa de discriminação de hipóteses permite, através da utilização de observações adicionais, uma melhor seleção das hipóteses.

Observando-se o grafo causal é possível selecionar um conjunto mínimo de anomalias para observação. Por exemplo, na Figura 118 se forem selecionadas as observações On1 e Oo2 é possível detectar sintomas quando ocorre qualquer uma das anomalias descritas no grafo.

Assim que for detectado algum sintoma poderia ser possível, para um sistema de diagnóstico que opere no modo ativo, requisitar informações adicionais a respeito dos seus contribuintes para possibilitar uma melhor precisão do diagnóstico.

Existe um problema em relação às observações defasadas em até 2 ciclos. Seu tempo de estabilidade de informação é alto quando comparado aos outros tipos de observação. Uma alternativa para contornar este problema seria diminuir o período de amostragem. Existe um limite mínimo, que depende de cada objeto gerenciado,

para o período de amostragem. Se o período for muito pequeno o objeto fica muito sensível a pequenos intervalos com alta “taxa” (geralmente este tipo de objetos medem taxas) causando falsos positivos, mesmo que na média medida em um intervalo maior de tempo a taxa seja aceitável.

Por este motivo, pode ser aconselhável a observação contínua de observações defasadas em até 2 ciclos.

8. Conclusão

Como já comentado anteriormente, é praticamente impossível a um operador acompanhar o comportamento de um ambiente computacional distribuído, mesmo com o apoio de plataformas de gerenciamento. É necessário sistemas de apoio que possam auxilia-lo na identificação dos problemas.

Porém, sistemas atemporais de correlação e diagnóstico são limitados à observações que ocorrem em um determinado instante ou conjunto de instantes, eliminando componentes temporais valiosos. Se por um lado simplifica o processo, por outro elimina uma informação adicional disponível.

O diagnóstico temporal permite a utilização de informações a respeito do momento de início, término e duração de determinadas ocorrências no sistema. Agora, não somente o estado ou valores, mas também a localização temporal destas ocorrências (os instantes de início e término bem como a duração) podem ser analisados, fornecendo um componente extra ao processo de seleção da melhor (ou melhores) hipótese.

Porém, de nada vale uma observação se a informação temporal nela contida for imprecisa. E, principalmente em redes de dados, esta imprecisão é intrínseca à dinâmica da observação, não sendo possível elimina-la. Cabe somente realizar seu modelamento de forma a explicitar tais imprecisões.

Este trabalho foi direcionado neste sentido: possibilitar o modelamento destas imprecisões temporais. Este é o primeiro passo. Muito existe ainda a ser percorrido. O diagnóstico temporal é, por um lado, valioso e, por outro, extremamente complexo. Por exemplo, uma observação recém chegada pode invalidar um diagnóstico do que ocorreu a uma hora atrás, forçando o sistema de diagnóstico a gerar uma nova solução a respeito da causa daquele problema. O espaço de dados cresce assim como a complexidade computacional, exigindo técnicas cada vez mais sofisticadas.

A possibilidade de relacionar causa e efeito no tempo, duração de eventos, dentre outros diversos aspectos tornam esta área muito atraente para novas pesquisas. Mas, se por um lado, na dimensão temporal muito existe ainda a percorrer, por outro surgiram diversos trabalhos relacionados ao modelamento de sistemas.

8.1 Conclusões

Este trabalho mostrou como é complexa a forma de obtenção de informações em um sistema distribuído. Na literatura, não foram encontrados termos e elementos que possibilitassem tal modelamento, fazendo necessária a criação de novos termos e propriedades associadas às observações nestes sistemas.

Mostrou também que é possível modelar as observações de um sistema distribuído e formata-las de modo adequado, explicitando suas imprecisões. Foi apresentada uma técnica de modelamento que pode ser utilizada por um sistema de diagnóstico temporal.

Para mostrar como esta informação pode ser útil em um processo de diagnóstico foi proposto um sistema muito simples de diagnóstico temporal (não variante no tempo) que utiliza, da dimensão temporal, somente os instantes de início e término de uma ocorrência. Esta informação adicional foi utilizada para aumentar a precisão do processo de diagnóstico.

Foi notado que, a dimensão temporal cria novas possibilidades de serem exploradas nesta classe de sistemas, e será tema de trabalhos futuros.

8.2 Contribuições

Foram várias as contribuições deste trabalho. Comecemos pelo modelamento temporal da observação, neste caso particular, a observação do comportamento de um sistema distribuído. Durante a formalização do processo de observação foi identificada a existência de uma entidade de fundamental importância para o modelamento e cuja presença na literatura é desconhecida. Foi necessário criar um termo para ela: “**objeto intermediário**” (denominação utilizada em contrapartida a

objeto gerenciado). A partir desse momento foi possível formalizar . alguns tipos de observação: **não defasada, defasada em até 1 ciclo e defasada em até 2 ciclos.** Também ficaram explícitos alguns intervalos de tempo, associados à observação, no qual não era possível determinar o estado de um objeto gerenciado. Daí surgiu os intervalos **IPI (Intervalo de Possibilidade de Início de ocorrência de estado), IC (Intervalo de Certeza de ocorrência de estado) e IPT (Intervalo de Término de ocorrência de estado)**. Dando prosseguimento ao processo de modelagem ficou claro que também existem **intervalos de incerteza (II)** nos instantes iniciais do processo de diagnóstico e próximo ao instante corrente. Em um intervalo de incerteza não é possível determinar o estado exato do objeto gerenciado. Os intervalos de incerteza estão também presentes quando ocorre perda de amostragem. Particularmente, em relação a este tópico, ficou claro o impacto da perda de amostragens em observações defasadas em até 2 ciclos.

Finalizado o processo de modelagem, foi proposto um **algoritmo que possibilita gerar os intervalos de certeza, possibilidade e incerteza** representativos, pela visão do gerente de monitoração, do estado do objeto gerenciado.

Também fez parte deste trabalho a **caracterização dos sistemas de diagnóstico para ambiente distribuído quanto ao tipo de observação utilizada**. A forma com que o sistema de diagnóstico interage com o ambiente para obtenção de observações causa impacto nos métodos de diagnóstico.

A utilização da informação sobre a localização temporal de uma ocorrência para correlacionamento com outra ocorrência levou a necessidade de agrupar os intervalos associados à uma determinada ocorrência. Este agrupamento foi denominado ***cluster***. Além disso foi definido o **operador "Interseção entre clusters"** que é um operador primitivo para a tarefa de diagnóstico.

Outra contribuição que é possível citar foi a proposição de **modelos reusáveis para representação de um sistema distribuído** (BERNAL, 1999b).

Por fim, vários sistemas de diagnóstico utilizam o grafo causal para representar o encadeamento de relações causa-efeito. **No grafo causal, foi explicitado o plano de observações em contrapartida ao plano de anomalias.** Estes geralmente são exibidos e representados em um mesmo plano como se fossem similares. Não são! A

utilização explícita em planos distintos facilita a compreensão do relacionamento e papel de cada um. Como exemplo desta “confusão” é possível citar sintomas (observações) como causa de outros sintomas (observações).

8.3 Limitações

Foi apresentado um estudo de caso simplificado de um sistema de diagnóstico temporal que utiliza as observações modeladas na forma de intervalos.

Entende-se que este trabalho seja o ponto de partida para trabalhos mais abrangentes a respeito de diagnóstico temporal e para estudo de técnicas de modelamento de observações em sistemas distribuídos.

Seria importante comparar, em um caso real, os resultados de um sistema de diagnóstico temporal com um sistema de diagnóstico para sistema distribuído, como por exemplo, SMARTS (KLIGER, 1995; OHIE, 1997a; OSHIE, 1997b; BROADMAN, 2002; SMARTS, 2000).

8.4 Trabalhos futuros

Existem alguns trabalhos futuros que podem ser citados como continuidade a este trabalho:

- Modelamento da observação: formalização matemática de agrupamento de intervalos (*clusters*) e operações sobre tais agrupamentos;
- Modelamento de sistemas: utilização de modelos comportamentais (de funcionamento correto ou anômalo) que utilize a completude de estados (não somente os estados anômalos);
- Pesquisa na área de sistemas de diagnóstico temporal, com a análise de outros métodos de diagnóstico temporal;
- Comparação efetiva de sistemas de diagnóstico atemporais com sistemas temporais em relação a velocidade e precisão de diagnóstico;

- Implementação de sistemas de diagnóstico em sistemas de produção, como é o caso do cluster iPAD (BERNAL, 1999a), que utilizaria este sistema para diagnóstico de anomalias.

Anexo 1. GERENCIAMENTO DE REDES

Gerenciar uma determinada entidade significa monitorar e controlar sua operação. Inicialmente os protocolos de gerenciamento de rede tinham como objetivo principal gerenciar os elementos de rede, permitindo principalmente:

- O gerenciamento remoto dos elementos de rede;
- Uma interface padronizada para a interação com os elementos de rede.

Os protocolos de gerenciamento, apesar de serem inicialmente utilizados no gerenciamento dos elementos de rede, foram definidos de forma a possibilitar o gerenciamento de qualquer tipo de entidade ativa existente em um sistema de computação, como por exemplo, o sistema operacional e seus subsistemas, os componentes de *hardware* de um computador, os serviços de rede oferecidos por um sistema, as aplicações de um sistema de computação, etc. Por este motivo, o termo "gerenciamento de rede" não reflete exatamente a funcionalidade atual e o termo preferido é "gerenciamento integrado de sistemas".

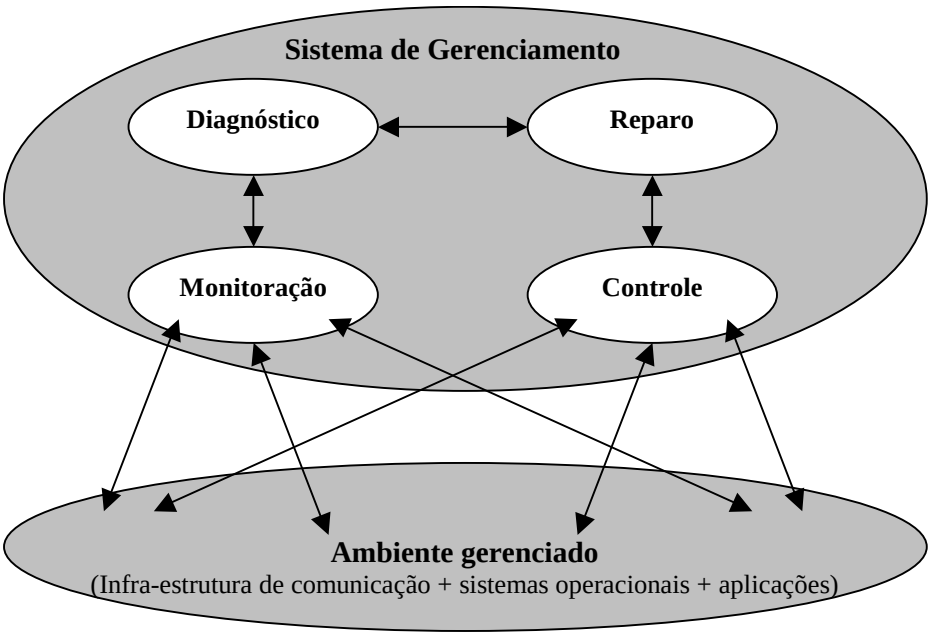


Figura 126 – Visão funcional geral de um sistema de gerenciamento

1 Áreas funcionais do gerenciamento

A *Network Management Forum* (NMF) da *International Organization for Standardization* (ISO) relacionou as seguintes áreas funcionais associadas ao gerenciamento (BRISA, 1993):

(i) **Gerenciamento de Falhas**: É o processo de localização de problemas (anomalias) no ambiente gerenciado. Envolve a descoberta, isolamento e correção do problema, se possível. Além disso, pode também atuar de forma pró-ativa, antecipando-se a uma falha, tipo de gerenciamento este mais desejado e importante. São atividades da gerência de falhas: identificar anomalias ocorridas no ambiente ou identificar tendências de ocorrência de anomalias; descobrir as causas destas anomalias; recomendar procedimentos de correção (ou reparo); e realizar e verificar os reparos efetuados;

(ii) **Gerenciamento de Configuração**: Envolve a configuração ou programação dos equipamentos para que estes atuem da forma estabelecida pela gerência, fornecendo subsídios para a preparação, partida, operação e suspensão dos serviços. Pode também tratar a descoberta das entidades do ambiente gerenciado, descoberta de sua configuração, descoberta de topologia;

(iii) **Gerenciamento de Contabilização**: Envolve a contabilização da utilização de recursos do ambiente gerenciado por determinadas entidades;

(iv) **Gerenciamento de Desempenho**: Envolve a medição da utilização de recursos do sistema, como por exemplo banda de comunicação, porcentagem de utilização e tempo de resposta, permitindo ao operador identificar situações de ociosidade ou de sobrecarga, além da eficiência das atividades realizadas;

(v) **Gerenciamento de Segurança**: Permite dar apoio à aplicação de políticas de segurança. Inclui funções para criar, controlar e eliminar mecanismos de segurança, registro e notificação de eventos de segurança. Diz respeito ao uso do gerenciamento de redes para monitorar e controlar mecanismos de segurança.

Estas são algumas áreas funcionais identificadas e especialmente importantes, que certamente não cobrem todo o escopo do gerenciamento integrado de sistemas.

2 Modelo de gerenciamento

Tais protocolos seguem o modelo gerente-agente. Neste modelo, apresentado na Figura 127, podem ser identificadas algumas entidades:

a) Objeto Gerenciado: Os objetos gerenciados são as entidades do sistema de computação passíveis de gerenciamento;

b) Agente: Módulo de *software* (usualmente um processo) responsável pela disponibilização das informações associadas a um ou mais objetos gerenciados (monitoramento) e pela atuação, mediante solicitação, sobre o objeto gerenciado (controle). O agente pode ainda transmitir notificações assíncronas sobre o comportamento de um objeto gerenciado. O agente é o responsável pela interação com os objetos gerenciados;

c) Gerente: Módulo de software responsável pela requisição de informações atualizadas sobre o comportamento dos objetos gerenciados e do controle sobre os objetos gerenciados. Também pode receber notificações assíncronas a respeito do comportamento de um objeto gerenciado. Para isto, interage com o agente utilizando-se de um protocolo de gerenciamento. Usualmente o gerente também disponibiliza uma interface ao operador;

d) MIB: A MIB (*Management Information Base*) é uma especificação das informações que podem ser trocadas entre o gerente e o agente. Isto possibilita que tais entidades possam (a) identificar precisamente o tipo de informação ou ação que está sendo requisitada (ou enviada) e (b) trocar tais informações;

e) Protocolo de Gerenciamento: especifica como é realizada a comunicação entre as entidades participantes do sistema de gerenciamento;

f) Operador: Responsável pela configuração do ambiente a ser gerenciado e por sua operação, verificando os alarmes recebidos, monitorando os dispositivos, etc.

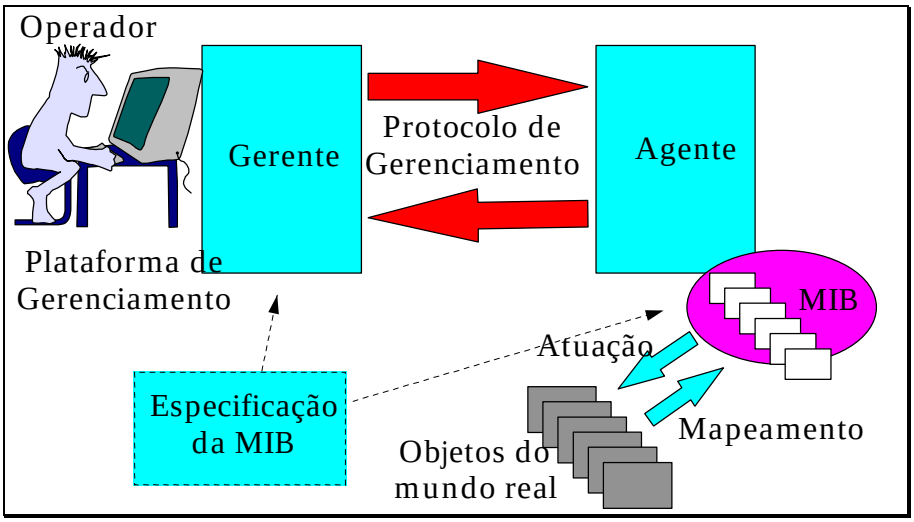


Figura 127. Modelo geral de gerenciamento

3 Protocolos de gerenciamento

Os dois principais protocolos de gerenciamento utilizados atualmente são o SNMP (RFC1155; RFC1157; RFC1212; RFC1212; RFC1213; RFC1214; RFC1215) (*Simple Network Management Protocol*) definido pelo IETF (*Internet Engineering Task Force*) e o CMIP (*Common Management Information Protocol*) definido pela ITU-T/ISO (BRISA, 1993). Mais recentemente foi estabelecido outro padrão, o DMI

(*Desktop Management Interface*) pelo DMTF (*Desktop Management Task Force*) voltado principalmente para gerenciamento de computadores.

As entidades de padronização não definem somente o protocolo de comunicação, mas também o modelo de informação (MIB) e outros aspectos para permitir o gerenciamento.

4 SNMP

Com a aceitação da tecnologia associada à Internet, o padrão SNMP tornou-se o mais popular no gerenciamento de redes de dados, com um papel também importante no gerenciamento de sistemas de telecomunicações.

4.1 SNMP v1

A versão 1 do protocolo (RFC1155; RFC1157; RFC1212; RFC1212; RFC1213; RFC1214; RFC1215) começou a ser definida a partir de 1989, e por ser simples, foi suportado por diversos fabricantes. Porém, esta versão possui problemas relacionados principalmente a segurança e forma de comunicação entre agente e gerente.

O protocolo SNMPv1 é composto pelas seguintes primitivas de serviço:

- **GetRequest**: Permite obter o valor de um ou mais objetos gerenciados;
- **GetNextRequest**: Permite obter o valor de um ou mais objetos gerenciados cuja instância de cada um é a próxima na ordem lexicográfica do objeto informado;
- **SetRequest**: Permite alterar o valor de um objeto gerenciado;
- **GetResponse**: Contém a resposta das requisições GetRequest, GetNextRequest ou SetRequest;
- **Trap**: Permite ao agente enviar uma notificação assíncrona ao gerente;

Para realizar o serviço de gerenciamento as entidades de serviço, neste caso os agentes e gerentes, trocam mensagens. O formato de cada mensagem SNMP está mostrado na Figura 128.

Figura 128 – Formato da mensagem SNMP

A Figura 129 mostra as principais formas de interação entre gerente e agente.

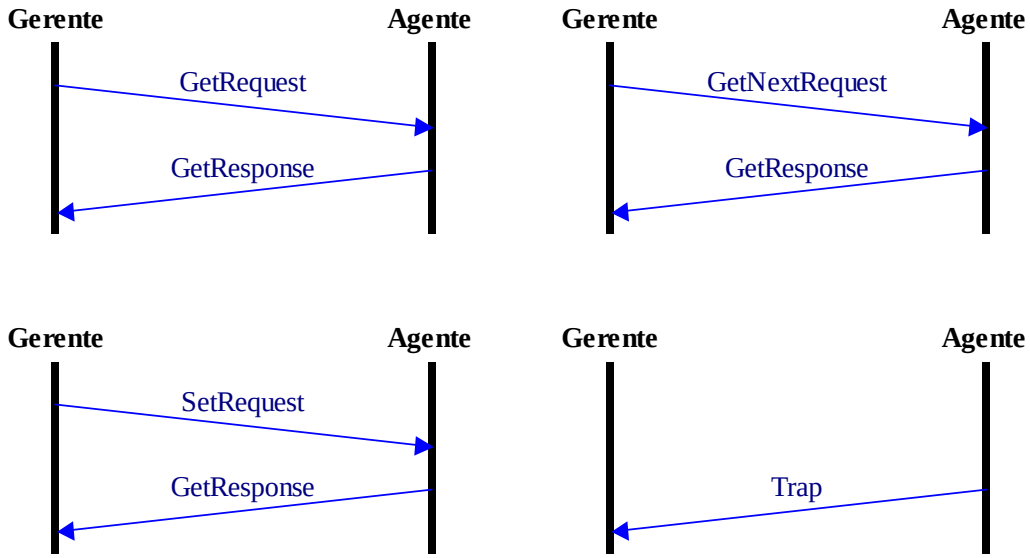


Figura 129 – Principais formas de interação entre gerente e agente.

4.2 SNMP v2

A partir de 1992 começaram estudos para uma nova versão. Por falta de consenso em relação aos aspectos de segurança resultou em várias versões: a versão *Party Based* (SNMPv2p) em 1992, a versão *User Based* (SNMPv2u e SNMPv2*) em 1996 e a versão *Community Based* (SNMPv2c) (RFC1901; RFC1902; RFC1903; RFC1904; RFC1905; RFC1906; RFC1907; RFC1908) em 1996, esta última a de maior aceitação.

4.3 SNMP v3

Em 1998 surgiu a padronização da versão 3, principalmente para resolver os problemas associados à segurança, permitindo a flexibilidade de acomodar os vários modelos de segurança (RFC2271; RFC2272; RFC2273; RFC2274; RFC2275).

4.4 Gerenciamento distribuído

Recentemente foram definidas algumas MIBs que permitem repassar tarefas de monitoração aos agentes. As principais MIBs desta classe são:

4.4.1 MIBS RMONx

As MIBs RMON (RFC1757, 1995), RMON2 (RFC2021, 1997) e SMON (RFC 2613, 1999) foram especialmente definidas para atuar como coletores (*probes*). A MIB RMON atua principalmente na monitoração de parâmetros da camada "interface" da pilha TCP/IP. A MIB RMON2 atua principalmente nas camadas superiores. A MIB SMON (RFC2613, 1999) estende as funcionalidades da MIB RMON.

4.4.2 "Expression MIB" e "Event MIB"

A "Expression MIB" (RFC2982, 2000) foi elaborada a partir do grupo de trabalho em gerenciamento distribuído do IETF. Ela permite definir tarefas de monitoração aos agentes de monitoração. É possível definir expressões envolvendo outros objetos gerenciados, distribuindo assim parte da tarefa de monitoramento. A "Event MIB" (RFC2981, 2000) permite a geração de eventos quando for transposto um determinado limiar.

4.4.3 "Script MIB"

Esta outra MIB possibilita ativar *scripts* nos agentes SNMP possibilitando a monitoração de entidades que não possuam agentes SNMP de gerenciamento.

4.4.4 “Ping”, “traceroute” e “nslookup” remoto

Outra MIB (RFC2925, 2000) permite a ativação de comandos “ping”, “traceroute” e “nslookup” diretamente na máquina gerenciada.

5 DMI

O padrão DMI define uma interface que disponibiliza informações sobre os componentes do sistema, sejam componentes de *software* ou de *hardware*.

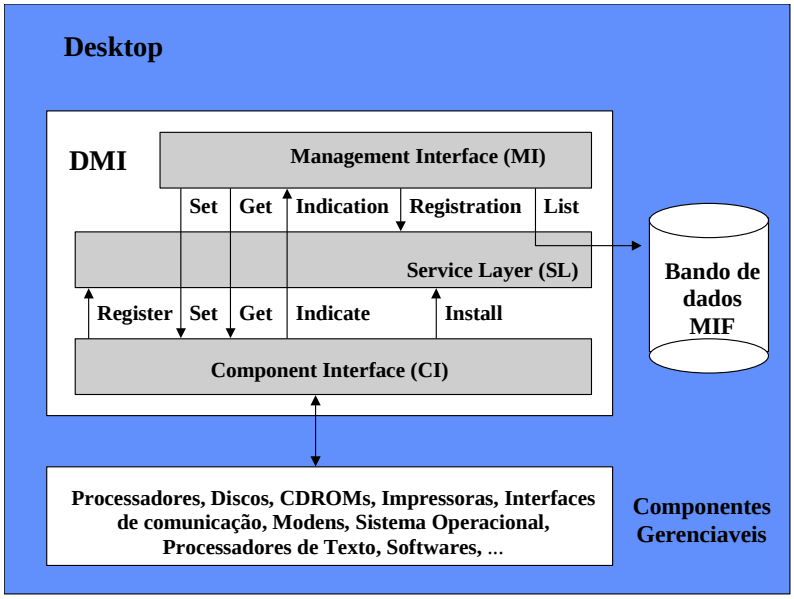


Figura 130. Componentes de um agente DMI (GHETIE, 1998)

O padrão DMI não define um protocolo de comunicação remoto. Define somente uma interface pela qual podem ser monitorados e controlados por um processo local. Este padrão é interessante pois define uma interface com o componente gerenciado, o que é especialmente importante para disponibilização de informações pelos componentes de *hardware*.

6 Gerenciamento OSI

O documento ISO 10.040 (CCITT X.701) descreve os termos, modelo geral de gerenciamento e requisitos para gerenciamento (SORTICA, 1999) (BRISA, 1993). O modelo geral de gerenciamento está mostrado na Figura 131.

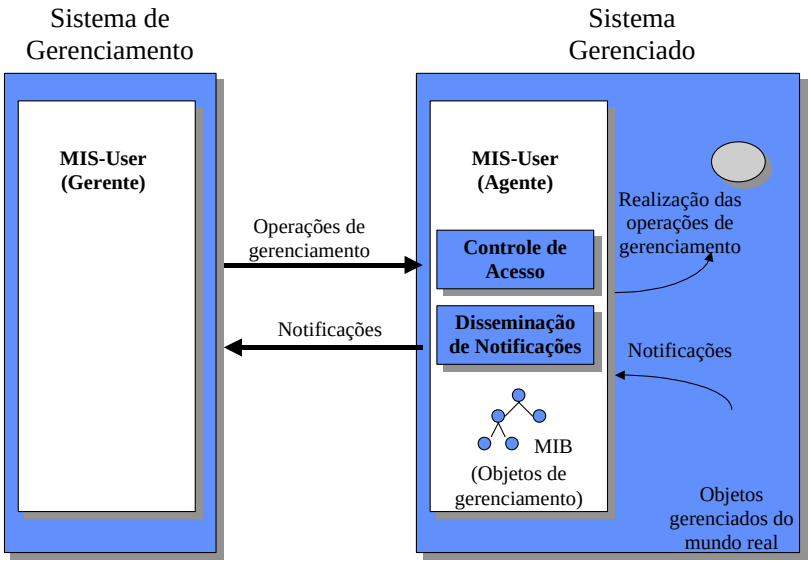


Figura 131. Interação entre gerentes, agentes e objetos gerenciados

Dentro deste contexto, o MIS-User é uma aplicação que faz uso dos serviços de gerenciamento e pode desempenhar tanto o papel de agente como de gerente (BRISA, 1993). Para uma determinada associação de gerenciamento, cada uma das entidades parceiras pode ter um dos dois papeis:

a) Gerente: um MIS-User que faz o papel de gerente é definido como parte de uma aplicação de gerenciamento de uma rede distribuída;

b) Agente: um MIS-User que faz o papel de agente é o responsável pela execução das operações de gerenciamento sobre os objetos gerenciados (entidades do mundo real) quando requisitado pelo gerente, e também por enviar eventos (notificações) que ocorreram na associação com os objetos gerenciados.

Os papeis (agente ou gerente) não são permanentemente designados aos MIS-Users. Eles podem fazer a função de agente, a de gerente, ou ambas, porém em interações distintas.

6.1 Comunicação de gerenciamento entre sistemas

Toda a informação de gerenciamento (operação ou notificação) trocada entre o gerente e o agente é realizada através do *Common Management Information Service Element* (CMISE), como mostrado na Figura 132, utilizando o protocolo CMIP (*Common Management Information Protocol*). O CMISE oferece serviços, *Common Management Information Service* (CMIS) associados a operações de gerenciamento e notificações.

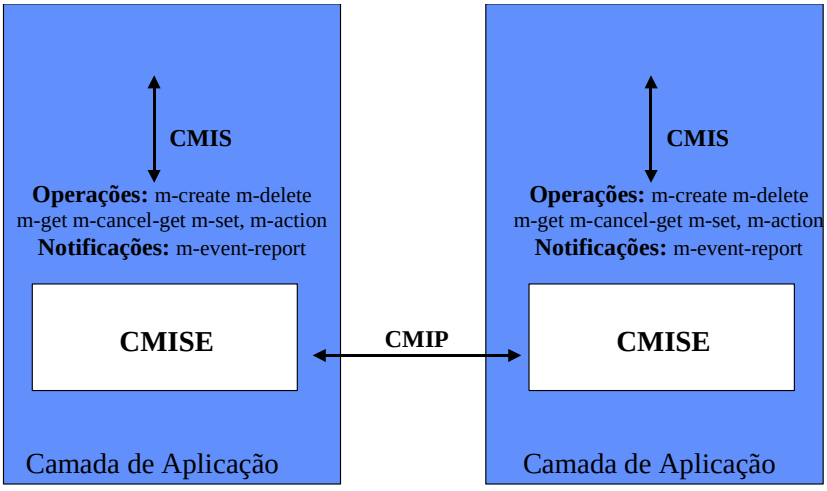


Figura 132. *Common Management Service Element*

Existe uma entidade da camada de aplicação, a SMAE (*Systems Management Application Entity*) que é utilizada pelas MIS-Users para se comunicar com as MIS-Users parceiras, como mostrado na Figura 133. A SMAE agrupa, além da CMISE, outras entidades funcionais importantes para o gerenciamento como a ACSE (*Association Control Service Element*) responsável pelo controle de associação (unidades funcionais existentes para verificar capacidades, versão de protocolo, controle de acesso, etc), a ROSE (*Remote Operation Service Element*) responsável pela transferência de dados, e a SMASE (*Systems Management Aplication Service Element*) que fornece os serviços de gerenciamento aos processos MIS-User.

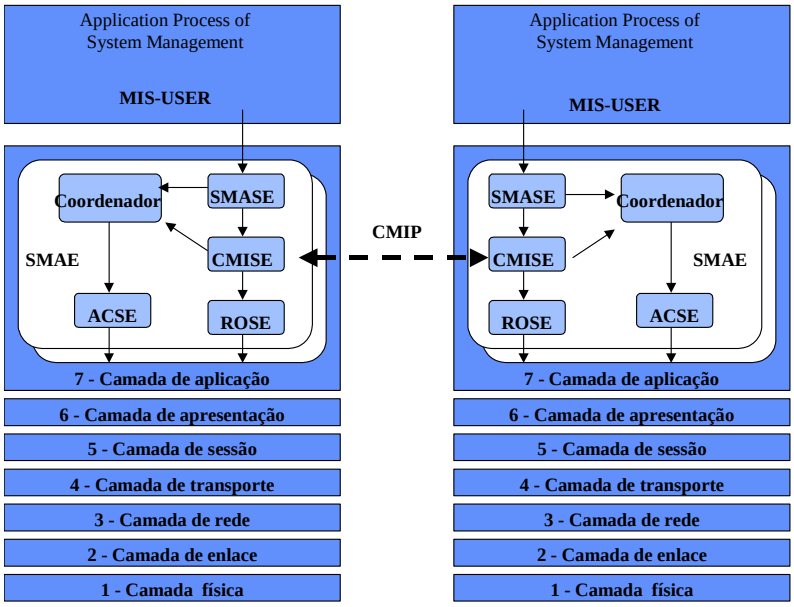


Figura 133. Pilha de protocolos no gerenciamento OSI (GHETIE, 1998).

6.2 Elementos de serviço do protocolo CMIP

O protocolo CMIP ("*Common Management Information Protocol*") possui as seguintes operações:

- **M-CREATE**: requisita ao agente para criar uma nova instância de um objeto gerenciado ou uma novo atributo em um objeto gerenciado;
- **M-DELETE**: requisita ao agente para remover uma instância de objeto gerenciado ou um atributo de um objeto gerenciado;
- **M-GET**: informa ao agente retornar valores de atributos de objetos gerenciados;
- **M-SET**: informa ao agente modificar valores de atributos de um determinado objeto gerenciado;
- **M-ACTION**: informa ao agente executar uma determinada ação no objeto gerenciado;
- **M-EVENT_REPORT**: enviado pelo agente para envio de informações aos gerentes;

6.3 Estrutura da informação de gerenciamento

Os conceitos básicos do modelo de informação usado pelos Sistemas de Gerenciamento OSI são definidos através da SMI (*Structure of Management Information*). O Modelo de informação definido é orientado a objetos. Assim, antes de mais nada é necessário definir cada classe de objeto. Isto consiste de definir o nome da classe, a superclasse ao qual pertence, seus atributos, as ações, o comportamento, os pacotes (grupo de atributos e ações), as operações suportadas em cada atributo e as possíveis notificações.

```
top MANAGED OBJECT CLASS
       CHARACTERIZED BY
                topPackage PACKAGE
                         BEHAVIOUR
                                  topBehaviour;
                         ATTRIBUTES
                                  objectClass        GET;
                                  nameBiding         GET;;;
       REGISTERED AS {smi2MObjectClass 14}

topBehaviour BEHAVIOUR
       DEFINED AS "... every managed object class is a
                specialization of either this generic class,
                top, or a specialization of subclass
                of top ..."
```

Figura 134. Exemplo da definição de uma classe de objeto.

6.3.5 Árvore de herança

Uma classe de objeto pode ser derivada de outra, definindo assim a relação de herança. Uma subclasse herda todas as propriedades de sua superclasse, de maneira irrestrita. Todas as classes devem ser derivadas da classe **top**, como ilustrado na Figura 135.

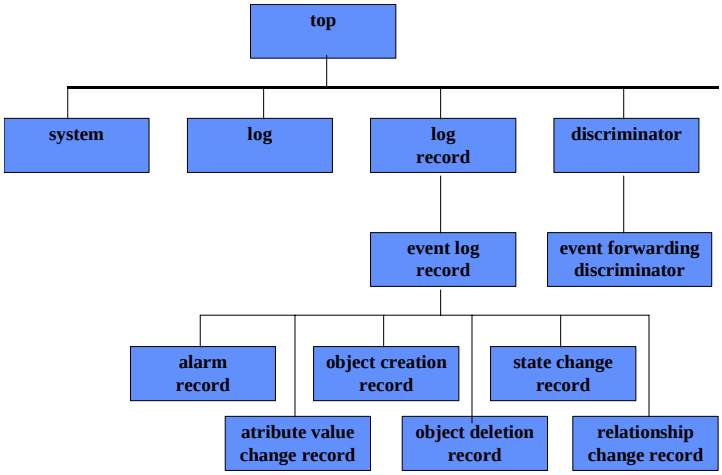


Figura 135. Exemplo de parte de uma árvore de herança.

É possível notar que as classes são utilizadas não somente para definição de objetos gerenciados, mas também para definição de objetos de informação associados ao gerenciamento como registro de *log*, e *Event Forwarding Discriminators* (EFD) (utilizados para definir as entidades que devem receber um determinado evento).

6.3.6 Árvore de nomeação

A árvore de herança não descreve o relacionamento entre os objetos (instâncias de objetos). Este relacionamento é descrito pela Árvore de Nomeação (ou também chamada de árvore *containment*). Nela, a existência de um objeto gerenciado é dependente da existência do objeto no qual ele está contido. Este relacionamento é definido pelo *name binding*. Um *name binding* indica qual o atributo que deve ser utilizado para identificação única do um objeto e qual a classe na qual ele pode estar contido. Este relacionamento forma a árvore de nomeação.

O nível mais alto desta hierarquia é chamado *root* (raiz), que é um objeto nulo e sempre existente. Supondo um determinado objeto desta árvore, todos os objetos subordinados (objetos contidos) são identificados por um nome característico relativo RDN (*Relative Distinguished Name*). Um RDN é formado por um atributo (chamado *distinguished attribute* e identificado pelo seu número de registro) e seu valor. A identificação de um objeto na árvore é realizada através do DN (*Distinguished Name*), também chamado de FDN (*Full Distinguished Name*). O DN é a seqüência de DN desde o objeto *root* até o objeto identificado.

Uma aplicação de gerenciamento possui uma visão dos objetos gerenciados do sistema em uma única arvore de nomeação. As operações de gerenciamento são realizadas sobre objetos gerenciados desta árvore.

6.3.7 Árvore de registo

As classes de objetos gerenciados, *actions*, *atributos* e todas as outras entidades do gerenciamento OSI possuem uma identificação única chamada número de registro. O número de registro é atribuído em função da posição do registro na árvore de registro.

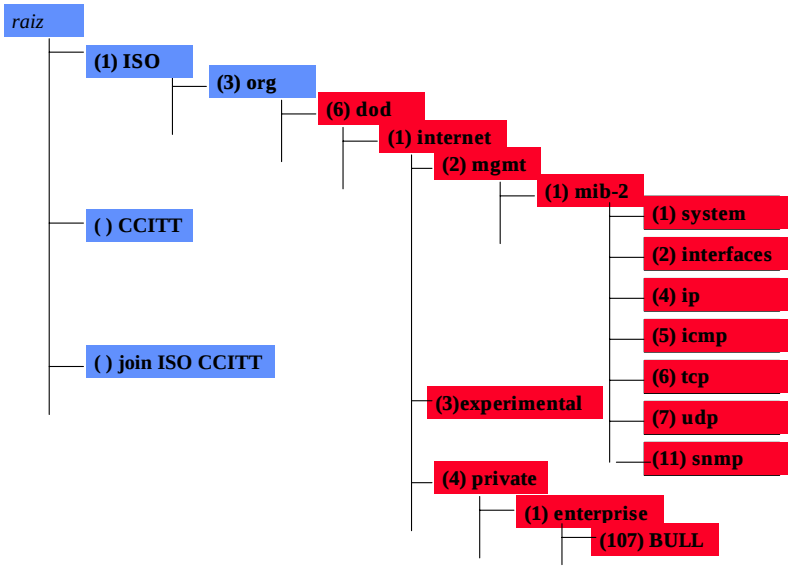


Figura 136. Árvore de registro.

6.3.8 Escopo

Em uma operação de gerenciamento é necessário identificar um objeto base que servirá como referência para a identificação dos objetos no qual deverão ser realizadas as operações de gerenciamento. Existem quatro tipos de escopo possíveis:

- Somente o objeto base;
- n-ésimo nível a partir do objeto base;
- Objeto base e todos os subordinados até (inclusive) os do n-ésimo nível;
- Objeto base e todos os seus subordinados (toda sub-árvore).

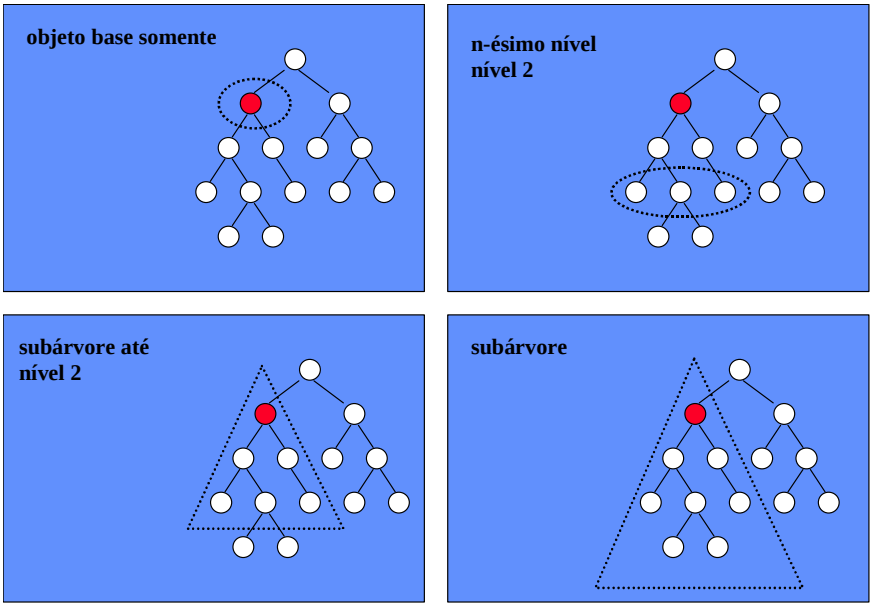


Figura 137. Exemplo de cada um dos quatro tipos possíveis de definição de escopo.

6.3.9 Filtro

Permite, em uma operação de gerenciamento, selecionar os objetos de acordo com expressões booleanas envolvendo a presença ou os valores de atributos de objetos.

7 TMN (Padrão OSI para telecomunicação)

Existem diversas tecnologias de rede de telecomunicações atualmente, dentre as quais pode-se destacar:

Tecnologia	**Serviços**
Telefone	Voz, dados
ISDN	dados
LP	dados
Novas Tecnologias de Rede *Broadband*	Internet de alta velocidade, Vídeo analógico, Vídeo Digital, *Pay Per View*, *Audio on demand*, *Near Video on Demand*, *Video on Demand*, Vídeo conferência

Uma rede de telecomunicação pode ser apresentada de forma rudimentar sendo constituída de rede principal, redes de acesso e de equipamentos terminais, como representado na Figura 138.

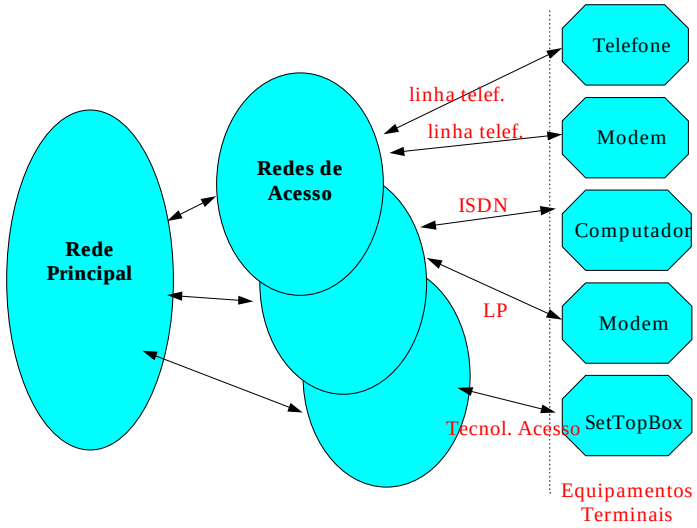


Figura 138. Exemplo de uma rede de telecomunicação

Algumas organizações estão atacando o problema das rápidas mudanças tecnológicas, de serviços e arquiteturas na indústria de telecomunicações e o processo lento de padronização internacional. Um exemplo destas mudanças é a necessidade atual por serviços e aplicações de vídeo conferência, áudio sob demanda e vídeo sob demanda. Cada vez mais, estas aplicações terão que operar em um ambiente multifornecedor e deverão também suportar características como interoperabilidade e flexibilidade de incorporação de novas funcionalidades. Para auxiliar na padronização de uma arquitetura especialmente voltada para o desenvolvimento de aplicações na área de telecomunicações foi fundado o TINA-C (*Telecommunications Information Network Architecture Consortium*), uma iniciativa de operadores, fornecedores de serviço e vendedores de produtos na área de telecomunicações. O objetivo de tal consórcio é a definição de uma arquitetura chamada TINA (*Telecommunications Information Network Architecture*) para a especificação de aplicações em telecomunicações (GAY, 1995). Esta arquitetura pode ser aplicada para redes de telecomunicações (banda larga, banda estreita) e redes de comunicação de dados.

A Tabela 5 mostra alguns tipos de serviços oferecidos por uma rede de telecomunicação.

Serviço	Descrição
Serviços de suporte	Transmissão entre dois pontos, incluindo roteamento e chaveamento (chaveamento de circuitos, chaveamento de pacotes) físico
Teleserviços	Inclui Todas capacidades para comunicação entre duas aplicações (telefone, computador)
Serviços básicos	Capacidade de manipular chamadas básicas (*call set-up, call release*)
Serviços suplementares	Capacidades opcionais que podem ser utilizadas para a suplementação de serviços básicos (*call forwarding, call waiting*)
Serviços de valor agregado	Serviços suplementares avançados. Serviços que são encapsulados, fornecidos e comercializados como produtos *stand-alone*: *Virtual Private Network Services, Video-on-Demand Services, Bandwidth-on-Demand Services, Security Services, QoS Services, ...*

Tabela 5 – Alguns serviços oferecidos por uma rede de telecomunicação.

7.1 Gerenciamento de redes de telecomunicações

Devido à falta ou à dificuldade de implementação de padrões relacionados à operação, manutenção e provisionamento para grande parte dos recursos gerenciados de uma rede de telecomunicação, a maior parte dos elementos de rede e dos equipamentos são lançados ao mercado com interfaces proprietárias (OMG, 1996). Nesta linha, vários equipamentos freqüentemente fazem uso de interfaces SNMP para seu gerenciamento, coexistindo na rede com equipamentos baseados no padrão ITU-T/OSI.

Seguindo o exemplo do modelo internet de gerenciamento (definido pela IETF), estão sendo estabelecidos consórcios industriais com o objetivo de desenvolvimento de padrões "*de facto*" ao invés de padrões "*de-jure*". Nesta área, a ITU-T/ISO é a responsável pelo estabelecimento de padrões "*de-jure*". O *Network Management Forum* (NMF), agora denominado *Tele Management Forum* (TMF), é uma das

organizações preocupadas no desenvolvimento de padrões "*de-facto*" especificamente para a indústria de telecomunicações, inclusive interagindo com a ITU-T/OSI. A *OMG Telecommunications Task Force* é outra organização deste tipo focada na padronização de interfaces baseadas em CORBA para a indústria de telecomunicações.

7.2 Padrão TMN

A padronização TMN (*Telecommunications Management Network*). (BRISA, 1993) (SORTICA, 1999) foi proposta pela antiga CCITT *(Consultative Committee for International Telegraph and Telephone*) agora denominada ITU-T *(International Telecommunications Union, Telecommunications. Standard Section*), descrita pelas recomendações series M.3000. Sua finalidade é fornecer uma arquitetura para gerenciamento de sistemas de telecomunicação e aumentar a interoperabilidade entre sistemas de gerenciamento. A arquitetura define uma rede lógica de comunicação de dados que permite a interconexão dos componentes do sistema de gerenciamento, dos dispositivos da rede de telecomunicacão e das demais entidades envolvidas em um sistema de telecomunicação. Esta rede lógica, distinta da rede de telecomunicações, pode se utilizar da infra-estrutura fornecida pela rede de telecomunicação, como ilustrado na Figura 139.

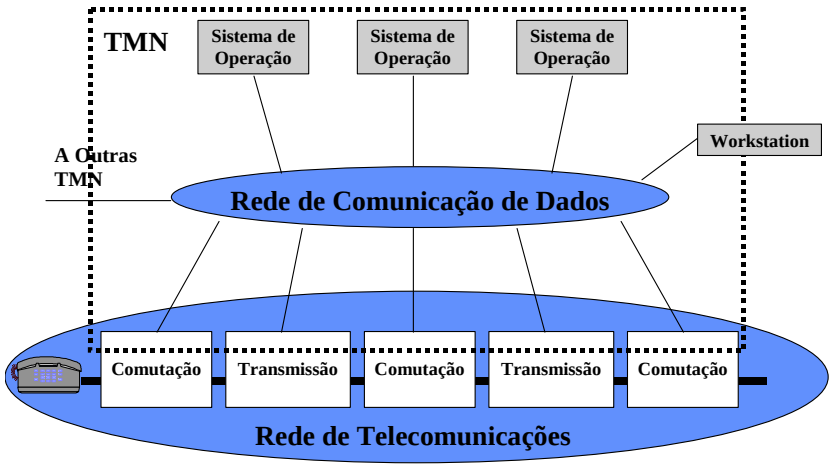


Figura 139. Relacionamento da TMN com a rede de telecomunicações

O modelo de referência OSI para gerenciamento de rede de telecomunicação define alguns blocos funcionais da TMN, descritos pela Tabela 6.

OSF	*Operations systems Function* - Sistema de suporte às operações.
MF	*Mediation Function* - Função de mediação
WSF	*Work Station Function* - Estação de trabalho
NEF	*Network Element Function* - Elemento de rede
QAF	*Q Adaptor Function* - Adaptador Q

Tabela 6 – Blocos funcionais da TMN.

O relacionamento destes blocos funcionais está mostrado na Figura 140.

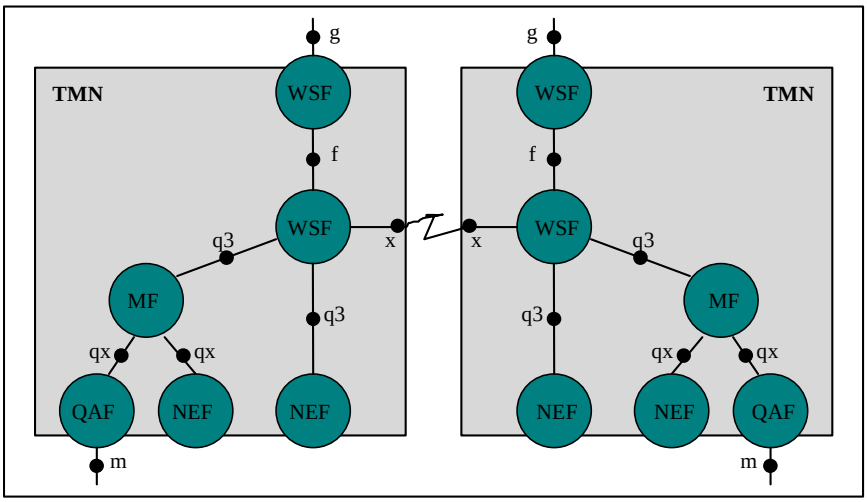


Figura 140. Inter-relacionamento entre os blocos funcionais da TMN

7.3 Modelo de camadas de gerenciamento

O gerenciamento de um sistema de telecomunicações pode ser também funcionalmente organizado em camadas, como mostrado na Figura 141.

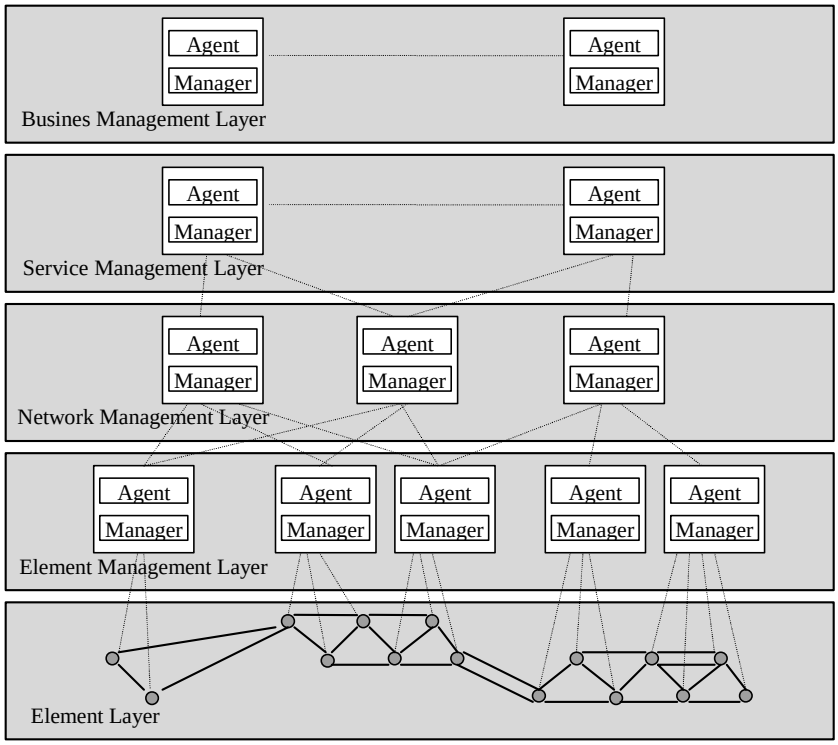


Figura 141. Camadas funcionais de suporte ao gerenciamento

A TMN divide em 5 camadas (BRISA, 1993):

- Camada de gerenciamento de negócios (*Busines Management Layer)*
- Camada de gerenciamento de serviços (*Service Management Layer*)
- Camada de gerenciamento de rede (*Network Management Layer*)
- Camada de gerenciamento de elemento de rede(*Element Management Layer*)
- Camada de elemento de rede (*Element Layer*)

7.3.10 Camada de elemento de rede

Corresponde aos componentes da rede de telecomunicações que necessitam ser gerenciados. Cada elemento de rede deve possuir agente para permitir seu gerenciamento.

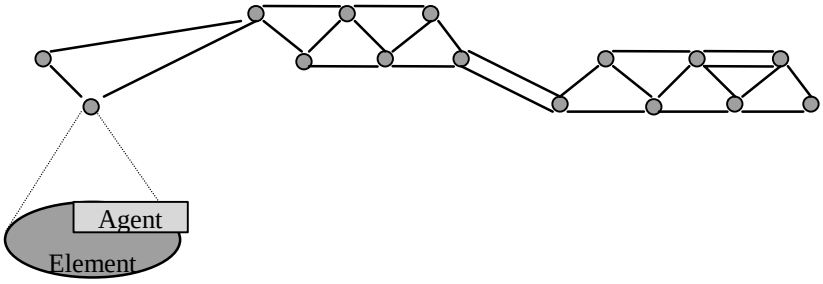


Figura 142. Um elemento de uma rede de telecomunicações

7.3.11 Camada de gerenciamento de elemento de rede

Composta por sistemas diretamente relacionados às atividades de gerenciamento individual dos elementos de rede, tais como supervisão, monitoração e controle de uma central telefônica ou de um sistema de transmissão e coleta de dados de desempenho de bilhetagem fornecidos pelos elementos de rede.

7.3.12 Camada de gerenciamento de rede

É a primeira camada que relaciona os elementos de rede individuais, possibilitando a visão da rede como um todo. É composta pelos sistemas destinados à operação, administração e manutenção de rede, tais como re-roteamento, planos de contingência, provisionamento de facilidades, detecção e isolamento de falhas.

7.3.13 Camada de gerenciamento de serviços

Composta por sistemas destinados à operação, administração e manutenção de serviços, abrangendo cadastro de usuários, relacionamento com usuários, provisionamento e manutenção de serviços, informações de faturamento, entre outros serviços.

7.3.14 Camada de gerenciamento de negócios

Composta por sistemas necessários ao gerenciamento do empreendimento como um todo, tais como atividades de controle e acompanhamento das metas e objetivos empresariais, planejamento estratégico e da expansão da planta, e análises gerenciais.

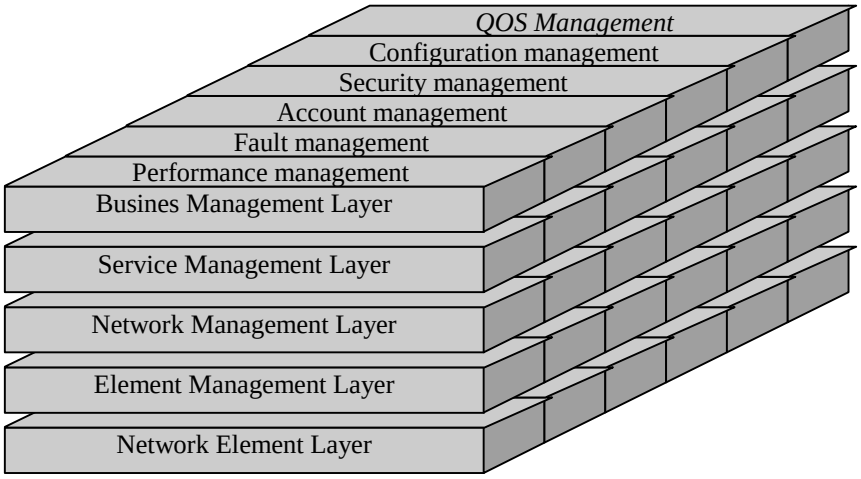


Figura 143. Planos de gerenciamento e as camadas funcionais

Anexo 2. RELAÇÕES CAUSAIS

Um dos relacionamentos mais importantes no diagnóstico de um sistema é causalidade, sendo a base de diversos sistemas de correlação e diagnóstico.

1 Relação causal

Será utilizada a notação *c* → *e* para denotar que "c" causa "e", sendo "c" denominado a causa e "e" denominado o efeito. A relação inversa "*e* efeito de *c*" será denotada por *e* ← *c*. A Figura 144 ilustra estas relações.

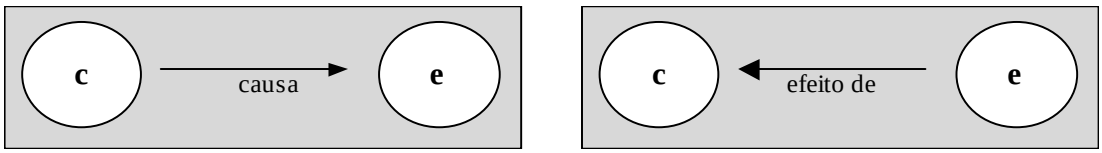


Figura 144 – Ilustração das relações "causa" e "efeito de".

A relação A→ S, no qual um sintoma S é causado por uma anomalia A, pode ser descrita por um grafo causal no qual os nós representam os eventos e arcos orientados representam a relação de causalidade. A Figura 145 mostra um exemplo de relacionamento causal entre anomalias e sintomas adaptado de (KLIGER, 1995).

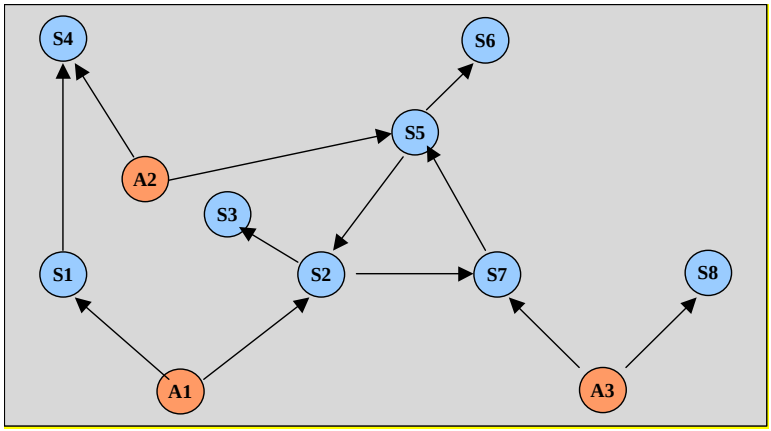


Figura 145 – Exemplo de grafo causal, adaptado de (KLIGER, 1995).

O relacionamento causal apresentado na Figura 145 foi extraído de um problema e apresenta algumas propriedades não desejáveis como, por exemplo, ciclos.

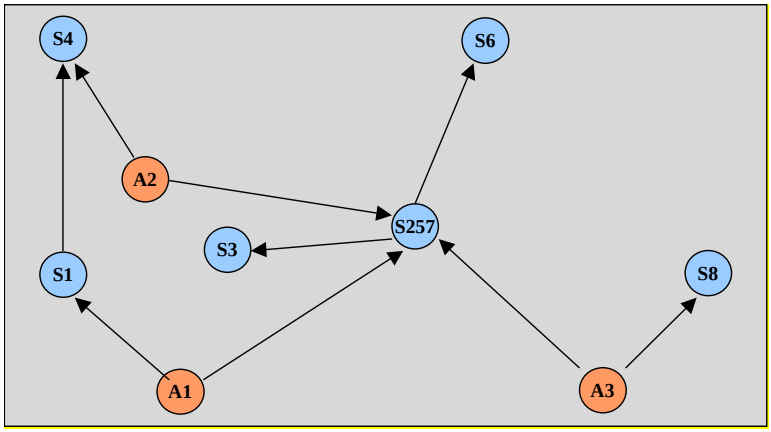


Figura 146 – Exemplo de grafo causal com a eliminação de ciclos.

A Figura 146 mostra um grafo causal derivado do anterior com a eliminação de ciclos e estados não desejados.

Formalmente, uma relação causal é um subconjunto do produto cartesiano de dois conjuntos.

Definição 22: Relação causal.

Seja C um conjunto das possíveis causas e "E" o conjunto dos possíveis efeitos. Então, é definida uma relação causal $\Re_C$ como sendo:

$\Re_C \subseteq C \times E$ tal que

$\Re_C = \{ (c,e) \mid c \text{ causa } e \}$

O conjunto C (causa), dependendo do caso, pode ser igual ao conjunto E (efeito). Por exemplo, nas relações causais entre anomalias tem-se $\Re_C \subseteq A \times A$, sendo A o conjunto de anomalias. Já nas relações causais entre anomalias e sintomas tem-se $\Re_C \subseteq A \times S$, sendo S o conjunto de sintomas que é distinto do conjunto de anomalias A.

Dentre as diversas propriedades, a relação causal é anti-simétrica e transitiva, sendo uma relação de ordem parcial.

2 Formas de representação da relação causal

Uma relação causal pode ser representada de diversas maneiras como: conjunto matemático, graficamente na forma de grafos, na forma de tabelas ou outros métodos. Por exemplo, seja a relação causal descrita na forma de conjunto matemático mostrado na Figura 147.

$$A = \{A1, A2, A3\}$$
$$S = \{S1, S2, S3, S4, S5, S6, S7, S8\}$$
$$\Re_C \subseteq A \times S$$
$$\Re_C = \{(A1,S2),(A1,S4),(A2,S2),(A2,S4),(A3,S2),(A3,S8)\}$$

Figura 147 – Exemplo de relação causal representada na forma de conjunto matemático.

Esta relação pode também ser representada na forma de grafo orientado, como mostrado na Figura 148. Neste caso, os vértices representam a união dos conjuntos A e S e as arestas indicam a relação causal entre as anomalias e sintomas.

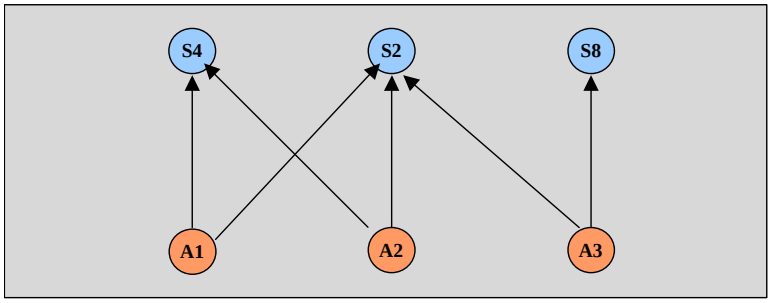


Figura 148 – Exemplo de relação causal representada graficamente na forma de grafo, adaptado de (KLIGER, 1995)

A relação pode ainda ser representada na forma de tabela. A Figura 149 mostra esta mesma relação causal representada na forma de tabela.

		A		
		A1	**A2**	**A3**
S	**S1**	0	0	0
	S2	1	1	1
	S3	0	0	0
	S4	1	1	0
	S5	0	0	0
	S6	0	0	0
	S7	0	0	0
	S8	0	1	1

Figura 149 – Exemplo de relação causal representada na forma de tabela.

3 Grafo de correlação

A técnica de correlação por livro-código é apoiada no grafo de correlação. O grafo de correlação é derivado de um grafo causal pela eliminação de sintomas indiretos e agregação de ciclos. A Figura 150 mostra o grafo de correlação derivado deste grafo causal.

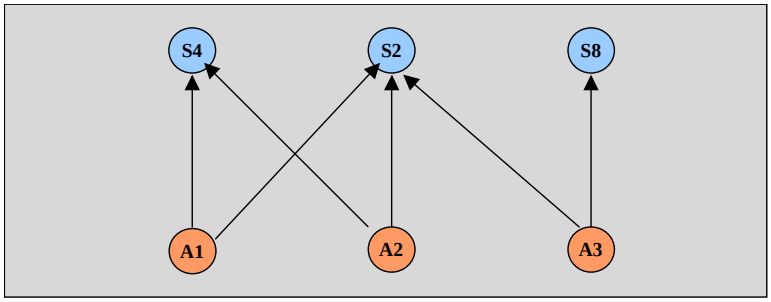


Figura 150 – Grafo de correlação derivado do grafo causal.

Matematicamente, isto é obtido através da utilização de um grafo de correlação bipartido. Para um determinado modelo de probabilidade causal <N,L,∅> é possível derivar um grafo de correlação N* correspondente ao grafo de causalidade N. Utilizando o operador ⊗ é possível associar uma medida de probabilidade a cada cadeia causal indo de uma anomalia A a um sintoma S, por exemplo A→ S1, S1→S2, S2→ S. A probabilidade de várias cadeias indo de A para S podem ser

combinadas utilizando o operador ⊕ a fim de fornecer a medida de probabilidade da correlação A ⇒ S.

4 Modelos de causalidade

É possível associar uma medida à causalidade. Assim sendo, podem existir diversos modelos de causalidade. Os exemplos mostrados anteriormente representam o modelo determinístico de causalidade. Antes de mais nada é necessário definir semi-anel.

Definição 23: Semi-anel.

Um semi-anel é um conjunto L parcialmente ordenado com ordem "≤" e dois operadores ⊗ e ⊕ tal que:

(i) <L, ⊗> é um semi-grupo com unidade **1**

(ii) <L, ⊕> é um semigrupo comutativo com uma unidade **0**

(iii) ∀ a,b ∈ L, a⊗b ≤ a,b e a,b ≤ a⊕b

(iv) ∀ a,b ∈ L, 0 ≤ a ≤ 1

Um semi-anel é utilizado para fornecer a medida associada a causalidade. Existem diversos tipos de modelos de causalidade que podem ser utilizados, como por exemplo:

- **Modelo determinístico de causalidade**. Este modelo utiliza um semi-anel L=**D**={0,1}, com ordem 0 ≤ 1 e operadores booleanos ⊗=∧ (*and*) com unidade 1 e ⊕=∨ (*or*) com unidade 0. O valor 1 indica possibilidade de causalidade e o valor 0 impossibilidade de causalidade;
- **Modelo probabilístico de causalidade**. Este modelo utiliza um semi-anel L=**P**=[0,1] com ordem numérica e operadores ⊗=* (*produto*) com unidade 1 e q1⊕q2=1-(1-q1)(1-q2) com unidade 0. O valor indica a probabilidade condicional do evento ocorrer;

- **Modelo temporal de causalidade**. Este modelo utiliza um semi-anel L=**T**=R* (conjunto dos números reais positivos) com ordem numérica inversa e operadores ⊗=+ (adição) com unidade 1 e operador ⊕=min (mínimo) com unidade ∞. O valor representa o tempo esperado para a causalidade ocorrer.

O modelo de probabilidade causal é uma tripla <N,L,∅> sendo N uma forma normal de grafo de causalidade, L é um semi-anel descrevendo o modelo de causalidade e ∅ é um mapeamento do conjunto de arestas de N em L associando uma medida de causalidade a cada implicação causal.

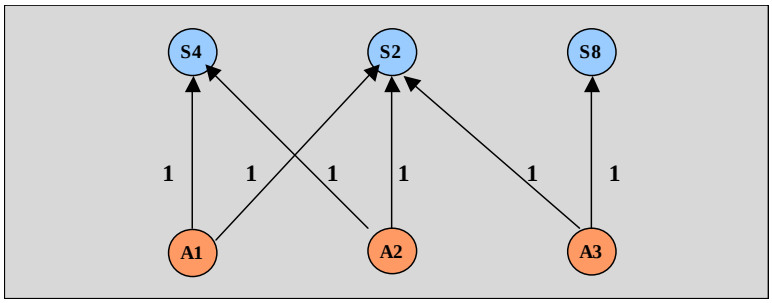


Figura 151 – Exemplo de modelo determinístico causal.

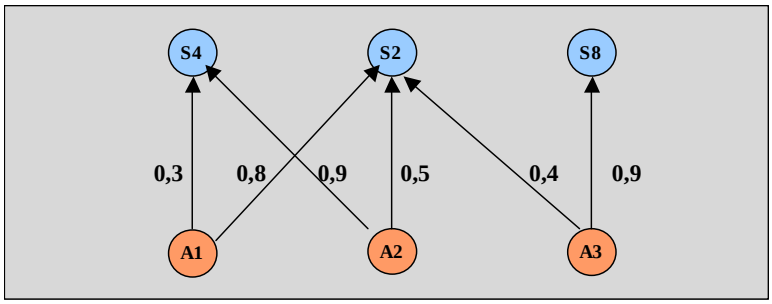


Figura 152 – Exemplo de modelo probabilístico causal.

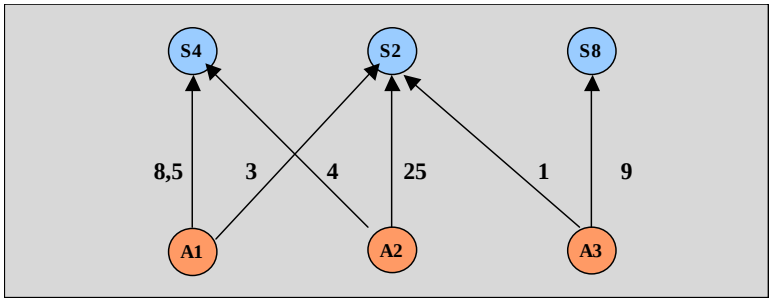


Figura 153 – Exemplo de modelo temporal causal.

Anexo 3. CÓDIGO DE HAMMING

1 Códigos de repetição

O meio mais intuitivo de possibilitar a detecção de erros (por exemplo, em transmissões ou armazenamento) é a utilização de códigos de repetição. Duplicando cada código original é possível detectar erros em um bit. Por exemplo, supondo que o código seja S1={0,1} o código com detecção de erros seria S1'={00,11}. Da mesma forma, se S2={000,001,010,011,100,101,110,111}, o código com detecção de erros seria S2'={000000,001001, 010010, 011011, 100100, 101101, 110110, 111111}.

2 Código de bloco binário

Um código de bloco binário de comprimento c dos quais d são bits de dados é denotado por código(c,d). No exemplo anterior S2' é um código(3*2,3).

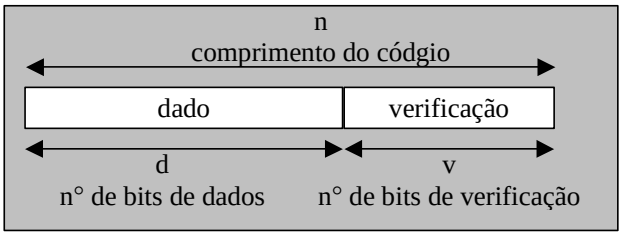


Figura 154 – Código de bloco binário

De maneira geral pode-se escrever que um código de bloco binário como:

```
código(d*s, d)
```

O número de bits com erro capaz de detecção é dado por:

```
(s – 1)
```

3 Redundância

Redundância é definido como sendo o número de bits de verificação utilizado dividido pelo mínimo necessário.

Genericamente, para um código de repetição (n,1) é possível:

- detectar $\lfloor n/2 \rfloor$ bits com erros
- corrigir $\lfloor (n-1)/2 \rfloor$ bits com erros

4 Syndrome

Syndrome é o nome dado aos bits adicionados para verificação. É possível definir uma expressão que informe o número de bits mínimo para detectar e indicar a localização de 1 bit trocado. Assim é possível saber quão eficiente é o método de detecção e correção de erros.

Suponha que sejam adicionados v bits de verificação. Com estes v bits é possível representar 2^v valores distintos. Estes 2^v valores distintos devem ser capazes de indicar:

- no caso de erro uma das n posições de erro
- ou que não existe erro

Portanto:

$$2^m >= n + 1 \quad , \text{ ou}$$
$$2^m >= d + v + 1$$

5 Código de Hamming

O código de Hamming é utilizado como código de redundância para detectar e recuperar erros em blocos de dados. A regra de Hamming é expressa pela s inequação mostrada na Figura 155.

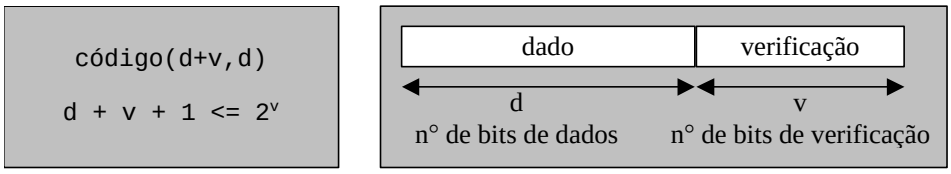


Figura 155 – (a) Regra de Hamming; (b) Palavra de código de Hamming

6 Distância do código de Hamming

A distância do código de Hamming entre duas palavras-código como sendo o número de dígitos no qual eles diferem. A distância é uma métrica satisfaz as seguintes condições:

- d(x,y) >= 0
- d(x,y) = 0 ⇔ x= y
- d(x,y) = d(y,x)
- d(x,y) <= d(x,y) + d(y,z)

Por exemplo:

- d(0,1)=1
- d(001,011)=1
- d(000,111)=3
- d(111,111)=0

A distância mínima de um código é o mínimo de todas as distâncias entre palavras-código distintas, ou seja:

```
distância_mínima(c) = min{ d(x,y) | x,y ∈C }
```

Anexo 4. REPRESENTAÇÃO DE TEMPO

Existem diversas formas possíveis de representação do tempo. Segundo Allen (1993), uma forma de representação geral que possa ser utilizada em diversas áreas da ciência deve possuir as seguintes características:

- A representação deve permitir imprecisão. Parte do "conhecimento temporal" é estritamente relativo e não associado a datas absolutas. Por exemplo: X ocorre antes que Y;
- Deve permitir a representação de "conhecimento incerto". Freqüentemente o "conhecimento temporal" não é conhecido completamente, possuindo apenas alguma restrição importante. Por exemplo: X não ocorre "ao mesmo tempo" que Y;
- A representação deve possibilitar a variação da granularidade da visão do tempo. Por exemplo: em história é costume considerar o tempo em termos de dias, meses ou anos, enquanto que em um sistema de computação, minutos, segundos, milisegundos ou até menos;
- O modelo deve suportar persistência. Por exemplo: "Se eu deixei meu carro no estacionamento de manhã ele ainda deve estar lá".

Porém, em áreas específicas, como em sistemas de tempo real (sendo possível citar outros como sistemas distribuídos), estas características, de acordo com Levi (1990), devem ser estendidas. A forma de representação de tempo para uso nesta classe de sistemas deve:

- Permitir a representação de eventos instantâneos, ou seja, pontos de tempo;
- Permitir a representação de eventos com duração, ou seja, intervalos de tempo;
- Suportar a descrição de eventos cuja duração pode ser contínua (convexa) ou não-contínua (não-convexa). Deve suportar também a representação da duração de eventos não-convexos periódicos e esporádicos;

- Permitir a construção de relações de ordem temporal;
- Suportar diversos níveis de granularidade, permitindo a resolução variar dependendo das necessidades;
- Suportar quantificação relativa e absoluta.

Particularmente, neste trabalho, serão utilizadas duas formas de representação de tempo: a representação baseada em pontos de tempo e a baseada em intervalos de tempo

Na representação baseada em pontos de tempo (*time-poind-based*) o sistema é definido como um conjunto de eventos que ocorrem em determinados instantes, com duração zero, e que resulta em uma troca de estado do sistema ou processo.

Na representação baseada em intervalos de tempo (*8*) o sistema é definido como um conjunto de atividades que consomem quantidades de tempo finita entre dois limites de tempo: início e fim.

A representação de tempo adotada no trabalho de (LEVI, 1990) é baseada no trabalho de (ALLEN, 1983). Ele estende a representação de intervalos de tempo convexo definida em (ALLEN, 1983) e a estende para intervalos de tempo não-convexos, convergindo para as necessidades práticas e teorias de alguns sistemas, como os sistemas de tempo real e os sistemas de diagnóstico. É possível, desta forma, trabalhar com: pontos de tempo, intervalos de tempo contínuo e intervalos de tempo não contínuo.

1 Ponto de Tempo

A representação baseada em ponto de tempo é a mais primitiva de todas. Decorre diretamente da álgebra tradicional, na qual cada instante de tempo está associado a um número real. Deste modo, todos os operadores e relações sobre números reais podem ser aplicados.

Definição 24: Ponto de tempo

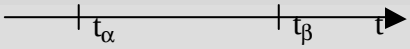
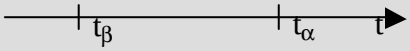
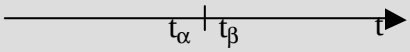
Um ponto de tempo (*time-point*) é um número real que representa o tempo de ocorrência de um evento instantâneo e é uma entidade indivisível. Um ponto de tempo está sempre associado a uma referência. Um ponto de tempo cujo valor seja zero define o instante de referência.

A ordem de dois pontos de tempo pode ser definida se, e somente se, estiverem associados à mesma referência. Deste momento em diante será assumido sempre um mesmo instante de referência.

1.1 Relações

Definição 25: Relações binárias entre pontos de tempo.

Sejam t_α e t_β pontos de tempo. É possível definir as seguintes relações de ordem:

antes	$t_\alpha < t_\beta$	
depois	$t_\alpha > t_\beta$	
simultâneo	$t_\alpha = t_\beta$	

2 Intervalo de Tempo Convexo

Apesar de ser mais direta a manipulação de pontos de tempo, devido principalmente à facilidade de manipulação de números reais e todos seus operadores e relações, inconscientemente nosso raciocínio em relação aos eventos geralmente é baseado na representação por intervalos de tempo. Normalmente, os únicos "tempos" que é possível identificar estão associados a ocorrências ou propriedades (ALLEN, 1984). Por exemplo, quando é dito "*... no instante que Pedro abriu a porta ...*" é possível identificar o seguinte momento: "*o instante que Pedro abriu a porta*". Mas também é possível observar mais atentamente esta ocorrência e decompô-la em diversos momentos: "*o instante que Pedro tocou a maçaneta*", "*o instante que a porta estava em movimento*" e "*o instante que Pedro largou a maçaneta*". Parece que sempre há

uma explicação causal mais detalhada do comportamento de cada instante. Uma boa analogia, então, seria aquela na qual os “tempos” correspondessem a intervalos sobre números reais.

Uma pergunta que pode surgir é: “Por que não aceitar também pontos de tempo?”. Primeiro, como será visto, não é necessário já que podem ser representados através de intervalos de tempo. Segundo, porque pontos de tempo instantâneos geram dificuldades semânticas sobre a lógica temporal. Se for aceito ponto de tempo, deveria ser também considerado se o intervalos de tempo fossem abertos ou fechados. Para ilustrar, seja o seguinte problema adaptado de (ALLEN, 1983):

> Seja C o intervalo de tempo de uma corrida e F o intervalo de tempo após a corrida. Seja P uma proposição representando o fato que a corrida está em andamento. Então P é verdadeira em C e ¬P é verdadeira em F. Assim, C e F devem “se encontrar em algum ponto”. Se ambos os extremos dos intervalos são abertos ou fechados, C e F devem ou compartilhar um ponto ou permitir entre eles um ponto. Desta forma existe um ponto de tempo no qual P ou P é verdadeiro ou existe um ponto de tempo no qual nem P nem ¬P seja verdadeiro. Uma solução para este problema é estipular, por convenção, que intervalos estão abertos no extremo anterior e fechados no extremos posterior. A artificialidade desta solução reforça o argumento contra os pontos de tempo. Por estes motivos, um ponto de tempo é considerado como um intervalo de tempo muito pequeno na representação por intervalo de tempo.

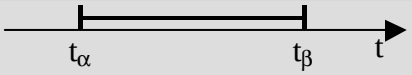
Apesar do trabalho de Levi (1990) se basear no trabalho de Allen (1983) ele despreza este possível problema semântico. Levi (1990) aplica esta teoria a sistemas de tempo real. Em um sistema de tempo real este problema é absorvido por outros dois: o problema da discretização do tempo e da coerência de relógio entre os diversos processadores. Em (LEVI, 1990) é apresentada uma solução para o problema de sincronização de relógio entre os equipamentos do sistema. Entretanto, estas afirmações de (LEVI, 1990) não são necessariamente válidas para sistemas computacionais de uso geral como a maior parte dos sistemas distribuídos nos quais a sincronização de relógio pode não existir. O problema da sincronização de tempo

entre equipamentos em um sistema distribuído é um problema prático real. sendo o tempo observado relativo a diferentes referências.

Definição 26: Intervalo de tempo convexo.

Um intervalo de tempo convexo $c = < t_\alpha , t_\beta >$ é um período de tempo contínuo no qual t_α é o ponto inicial e t_β é o ponto final:

$$c = < t_\alpha , t_\beta > \equiv \{t: t_\alpha \leq t \leq t_\beta\}$$



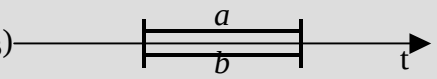
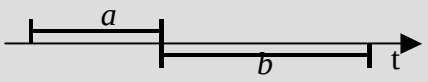
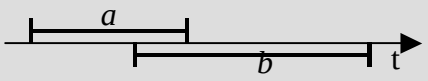
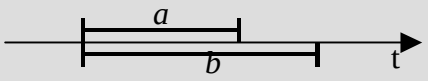
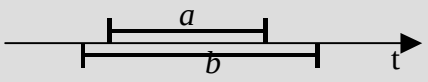
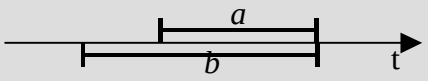
Um intervalo de tempo convexo, portanto, é um conjunto de pontos de tempo (delimitados pelos instantes t_α e t_β). Um ponto de tempo t_α pode, desta forma, ser representado pelo intervalo de tempo $c = < t_\alpha , t_\alpha >$. Neste caso, o conjunto possui somente um elemento, o próprio ponto t_α .

2.1 Relações primitivas entre intervalos de tempo convexo

É possível definir treze relações de ordem sobre intervalos de tempo convexos.

Definição 27: Relações de ordem binária sobre intervalos convexos.

Sejam *a* e *b* intervalos de tempo convexo tais que $a = < t^a_\alpha , t^a_\beta >$ e $b = < t^b_\alpha , t^b_\beta >$. É possível definir as seguintes relações de ordem:

Relação	Relação inversa		
igual $a = b$	igual $b = a$	$(t^a_\alpha = t^b_\alpha) \wedge (t^a_\beta = t^b_\beta)$	
precede $a < b$	sucede $b > a$	$t^a_\beta < t^b_\alpha$	
encontra $a \Uparrow b$	é-encontrado $b \Downarrow a$	$t^a_\beta = t^b_\alpha$	
sobrepõe $a \otimes b$	é-sobreposto $b \otimes$" a	$t^a_\alpha < t^b_\alpha < t^a_\beta < t^b_\beta$	
inicia $a \uparrow b$	é-iniciado $b \uparrow$" a	$t^a_\alpha = t^b_\alpha < t^a_\beta < t^b_\beta$	
durante $a \supset b$	contém $b \subset a$	$t^b_\alpha < t^a_\alpha < t^a_\beta < t^b_\beta$	
termina $a \downarrow b$	é-terminado $b \downarrow$" a	$t^b_\alpha < t^a_\alpha < t^a_\beta = t^b_\beta$	

2.2 Relações adicionais entre intervalos de tempo convexo

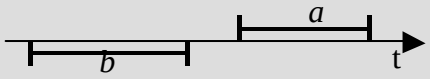
Além das relações de ordem, existem mais duas relações que podem ser definidas: disjunto e dentro.

Definição 28: Outras relações

Sejam *a* e *b* intervalos de tempo convexo tais que $a = < t^a_\alpha , t^a_\beta >$ e $b = < t^b_\alpha , t^b_\beta >$. É possível definir as seguintes relações:

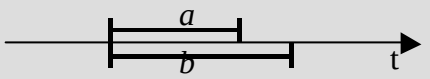
disjunto

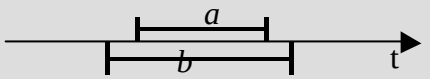
$a \nabla b \equiv (a < b) \vee (a > b)$
$\equiv (t^a_\beta < t^b_\alpha) \vee (t^a_\alpha < t^b_\beta)$

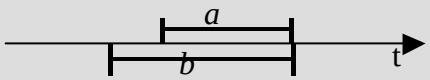


dentro

$a \subseteq b \equiv (a \uparrow b) \vee (a \subset b) \vee (a \downarrow b)$
$\equiv (t^a_\alpha \leq t^b_\alpha) \vee (t^a_\beta \leq t^b_\beta)$







2.3 Operadores sobre intervalos de tempo convexo

Também é possível definir alguns operadores sobre intervalos de tempo convexo, como: operador duração, operador interseção e operador cobertura.

Definição 29: Duração de um intervalo convexo.

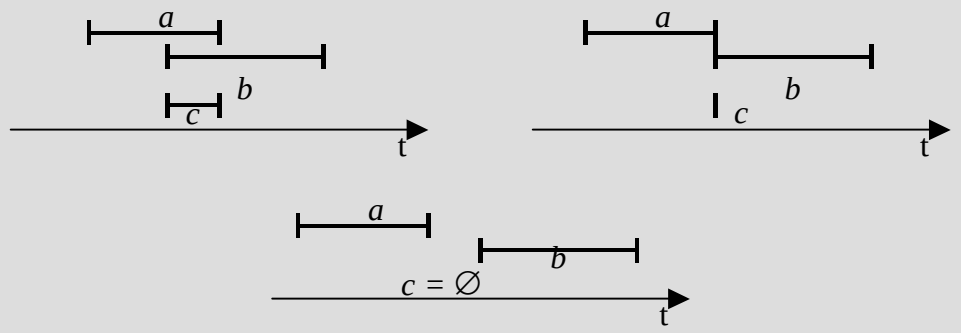
A duração de um intervalo de tempo convexo $c = <t_\alpha , t_\beta >$, representada por $\| c \|$, é dada por:

$$\| c \| = \| <t_\alpha , t_\beta > \| = | t_\alpha - t_\beta |$$

Definição 30: Interseção de intervalos convexos.

Sejam $a = < t^a_\alpha , t^a_\beta >$ e $b = < t^b_\alpha , t^b_\beta >$ dois intervalos de tempo convexos. A interseção dos intervalos *a* e *b* é:

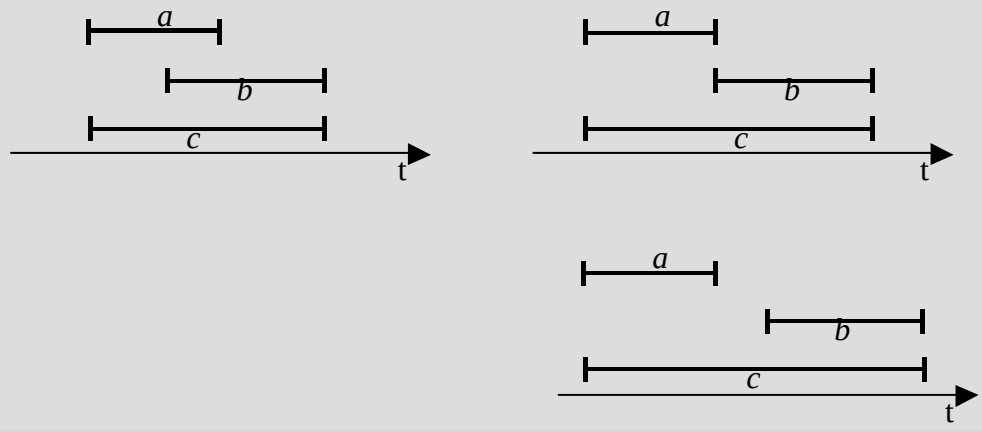
$$c = a \cap b \equiv \begin{cases} <\max(t^a_\alpha, t^b_\alpha), \min(t^a_\beta, t^b_\beta)> & \text{para } \neg((a < b) \vee (a > b)) \\ \varnothing & \text{para } (a < b) \vee (a > b) \end{cases}$$



Definição 31: Cobertura de intervalos convexos

A cobertura c de dois intervalos de tempo convexos $a = < t^a_\alpha , t^a_\beta >$ e $b = < t^b_\alpha , t^b_\beta >$ é o seguinte intervalo convexo:

$$c = a \cup b \equiv <\min(t^a_\alpha, t^b_\alpha), \max(t^a_\beta, t^b_\beta)>$$



2.4 Transitividade

As 13 relações de intervalo de tempo convexo apresentadas anteriormente podem expressar qualquer relação que possa ocorrer entre dois intervalos convexos quaisquer. Estas relações podem ser representadas através de um grafo, no qual os

nós representam os intervalos convexos e os arcos representam as possíveis relações entre os dois nós aos quais está conectado.

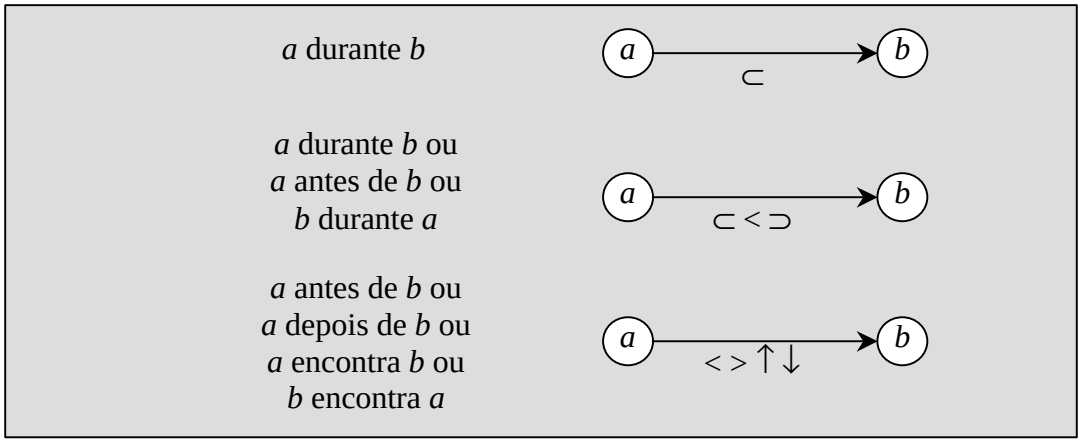


Figura 156 – Exemplo de representação de conhecimento temporal através de grafos.

A cada arco podem estar associadas todas as possíveis relações que possam ocorrer entre dois nós. Assim, cada arco pode conter de 1 até 13 relações.

Uma propriedade importante é a transitividade das relações temporais. A transitividade possibilita a realização de inferências em relação a intervalos de tempo não relacionados diretamente. Se $a\ \Re_1\ b$ e $b\ \Re_2\ c$ então é possível definir as possíveis relações entre *a* e *c* através do cálculo da transitividade. A Tabela 7 mostra a tabela de transitividade.

$a\ \mathfrak{R}_2\ c$		$b\ \mathfrak{R}_2\ c$												
		=	<	>	⊂	⊃	⊗	⊗”	⇑	⇓	↑	↑”	↓	↓”
$a\ \mathfrak{R}_1\ b$	=	=	<	>	⊂	⊃	⊗	⊗”	⇑	⇓	↑	↑”	↓	↓”
	<	<	<		< ⊂ ⊗ ⇑ ↑	<	<	< ⊂ ⊗ ⇑ ↑	<	< ⊂ ⊗ ⇑ ↑	<	<	< ⊂ ⊗ ⇑ ↑	<
	>	>		>	> ⊂ ⊗” ⇓ ↓	>	> ⊂ ⊗” ⇓ ↓	>	> ⊂ ⊗” ⇓ ↓	>	> ⊂ ⊗” ⇓ ↓	>	>	>
	⊂	⊂	<	>	⊂		< ⊂ ⊗ ⇑ ↑	> ⊂ ⊗” ⇓ ↓	<	>	⊂	> ⊂ ⊗” ⇓ ↓	⊂	< ⊂ ⊗ ⇑ ↑
	⊃	⊃	< ⊃ ⊗ ⇑ ↓”	< ⊃ ⊗” ⇑ ↑”	⊂ =⊃ ⊗ ⊗” ↑ ↑” ↓ ↓”	⊃	⊃ ⊗ ↓”	⊃ ⊗” ↑”	⊃ ⊗ ↓”	⊃ ⊗” ↑”	⊃ ⊗ ↓”	⊃	⊃ ⊗” ↑”	⊃
	⊗	⊗	<	> ⊃ ⊗” ⇓ ↑”	⊂ ⊗ ↑	< ⊂ ⊗ ⇑ ↓”	< ⊗ ⇑	⊂ =⊃ ⊗ ⊗” ↑ ↑” ↓ ↓”	<	⊃ ⊗” ↑”	⊗	⊃ ⊗ ↓”	⊂ ⊗ ↑	< ⊗ ⇑
	⊗”	⊗”	< ⊃ ⊗ ⇑ ↓”	>	⊂ ⊗” ↓	> ⊃ ⊗” ⇓ ↑”	⊂ =⊃ ⊗ ⊗” ↑ ↑” ↓ ↓”	> ⊗” ⇓	⊃ ⊗ ↓”	>	⊂ ⊗” ↓	> ⊗” ⇓	⊗”	⊃ ⊗ ↓”
	⇑	⇑	<	> ⊃ ⊗” ⇓ ↑”	⊂ ⊗ ↑	<	<	⊂ ⊗ ↑	<	= ↓ ↓”	⇑	⇑	⊂ ⊗ ↑	<
	⇓	⇓	< ⊃ ⊗ ⇑ ↓”	>	⊂ ⊗” ↓	>	⊂ ⊗” ↓	>	= ↑ ↑”	>	⊂ ⊗” ↓	>	⇓	⇓
	↑	↑	<	>	⊂	< ⊃ ⊗ ⇑ ↓”	< ⊗ ⇑	⊂ ⊗” ↓	<	⇓	↑	= ↓ ↓”	⊂	< ⊗ ⇑
	↑”	↑”	< ⊃ ⊗ ⇑ ↓”	>	⊂ ⊗” ↓	⊃	⊃ ⊗ ↓”	⊗”	⊃ ⊗ ↓”	⇓	= ↓ ↓”	↓”	⊗”	⊃
	↓	↓	<	>	⊂	> ⊃ ⊗” ⇓ ↑”	⊂ ⊗ ↑	> ⊗” ⇓	⇑	>	⊂	> ⊗” ⇓	↓	= ↓ ↓”
	↓”	↓”	<	> ⊃ ⊗” ⇓ ↑”	⊂ ⊗ ↑	⊃	⊗	⊃ ⊗” ↑”	⇑	⊃ ⊗” ↑”	⊗	⊃	= ↓ ↓”	↓”

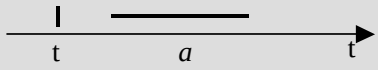
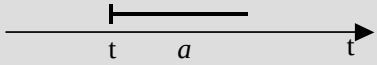
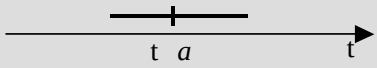
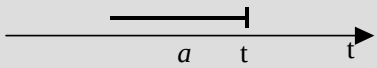
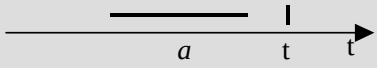
Tabela 7 – Tabela de transitividade de relações, extraída de (ALLEN, 1993).

3 Relações entre intervalos e pontos de tempo

É possível também definir algumas relações entre um ponto de tempo e um intervalo de tempo e entre um intervalo de tempo e um ponto de tempo.

Definição 32: Relações de ordem binária entre pontos de tempo e intervalos convexos.

Seja t um ponto de tempo e a = < t_α , t_β > um intervalos de tempo convexo. É possível definir as seguintes relações de ordem:

Relação	**Relação inversa**		
antes t < *a*	depois t > *a*	(t < t_α)	
inicia t ↑ *a*	é-iniciada t_α ↑" *a*	(t = t_α)	
durante t ⊃ *a*	contém t ⊂ *a*	(t_α < t < t_β)	
termina t ↓ *a*	é-terminada t ↓" *a*	(t = t_β)	
depois t > *a*	antes t < *a*	(t > t_β)	

4 Conclusão

As relações temporais são fundamentais para expressar o conhecimento temporal e realização do raciocínio temporal. Foram descritas relações temporais qualitativas:

- 3 relações primitivas entre dois pontos de tempo;
- 13 relações primitivas entre dois intervalos de tempo;
- 2 relações adicionais sobre intervalos de tempo;
- 5 relações primitivas entre intervalo de tempo e ponto de tempo;
- 5 relações primitivas entre ponto de tempo e intervalo de tempo.

Anexo 5. EXEMPLOS DE MODELAMENTO DE OBSERVAÇÃO

A seguir são mostrados alguns exemplos de modelagem de observação, utilizando os algoritmos descritos no capítulo 6.

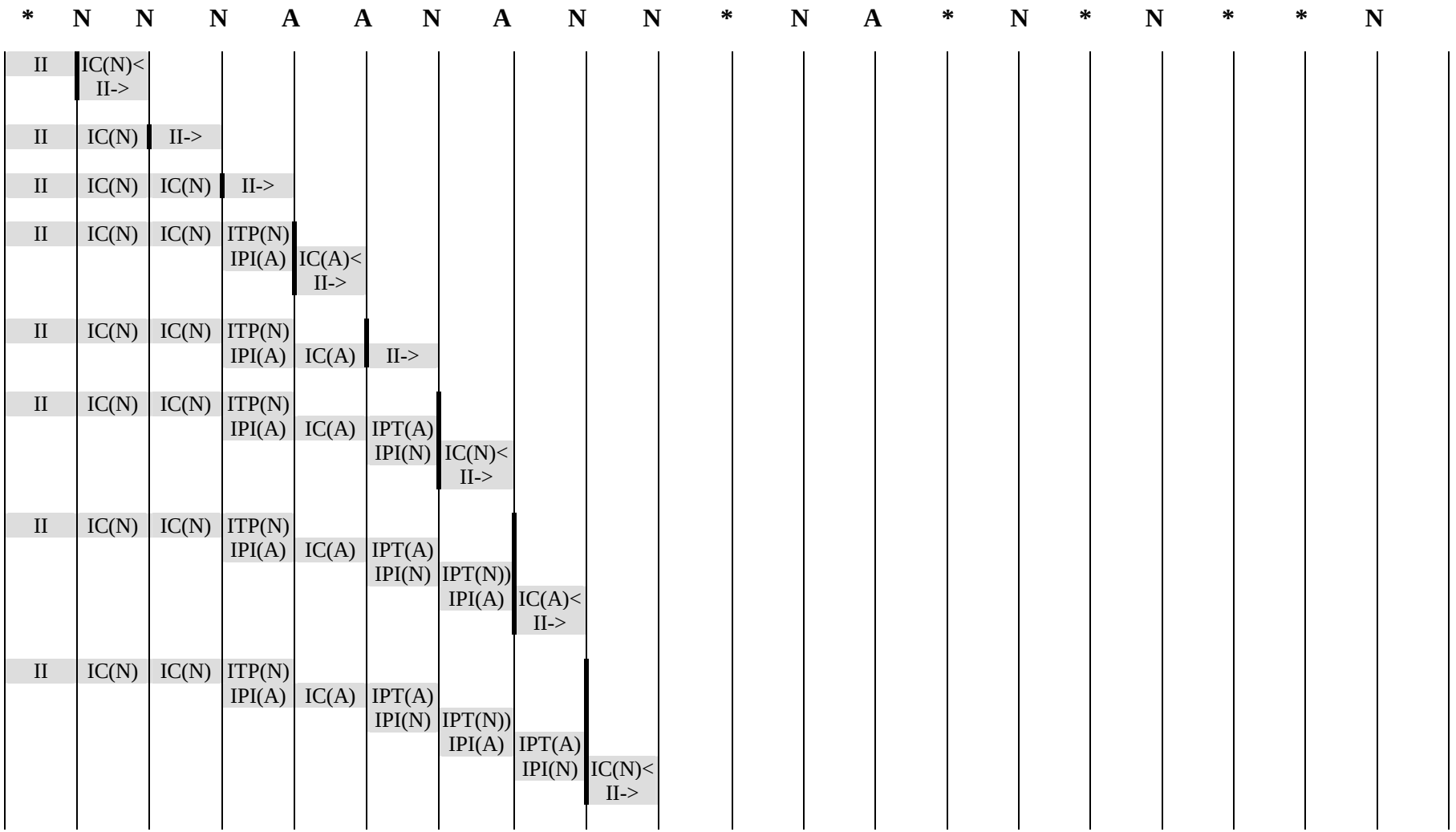
1 Exemplo #1 – Defasada em até 1 ciclo

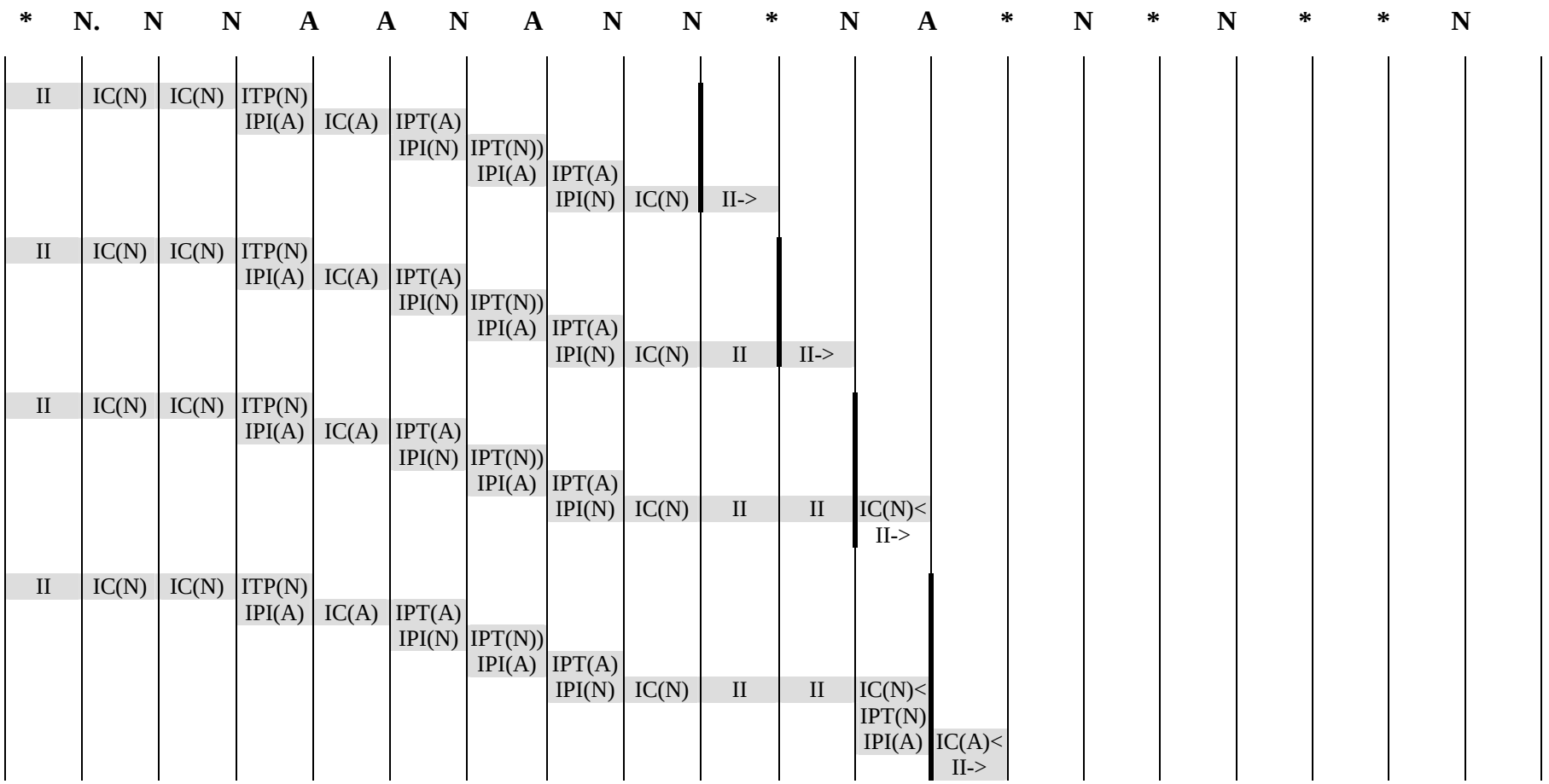
A figura a seguir mostra um exemplo da evolução do modelamento em intervalos de tempo a partir de uma seqüência de estados observados de objetos intermediários defasados em até 1 ciclo. A seqüência de estados simulada é:

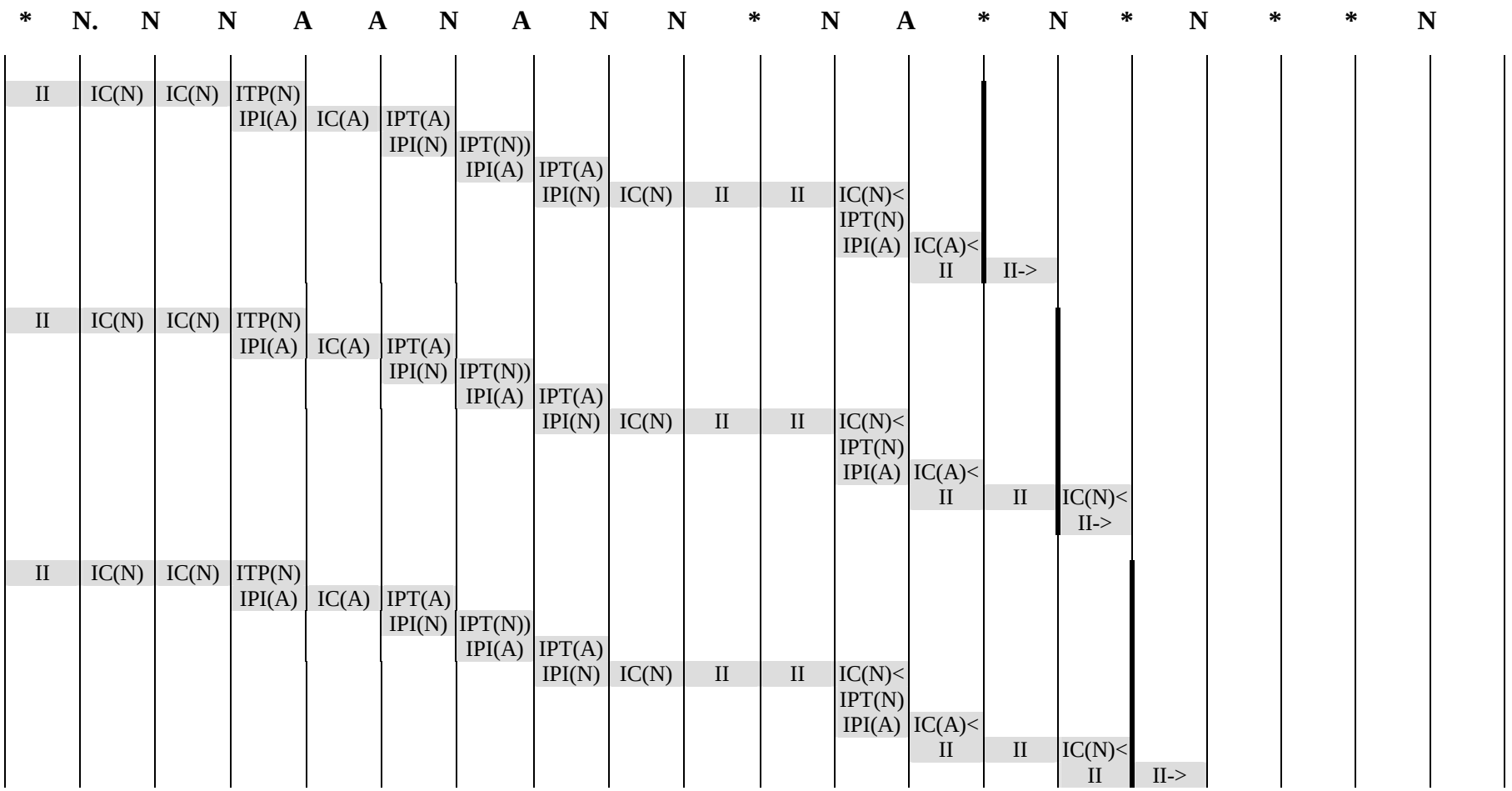
```
Seqüência de estados:
    N, N, A, A, N, A, N, N, *, N, A, *, N, *, N, *, *, N
```

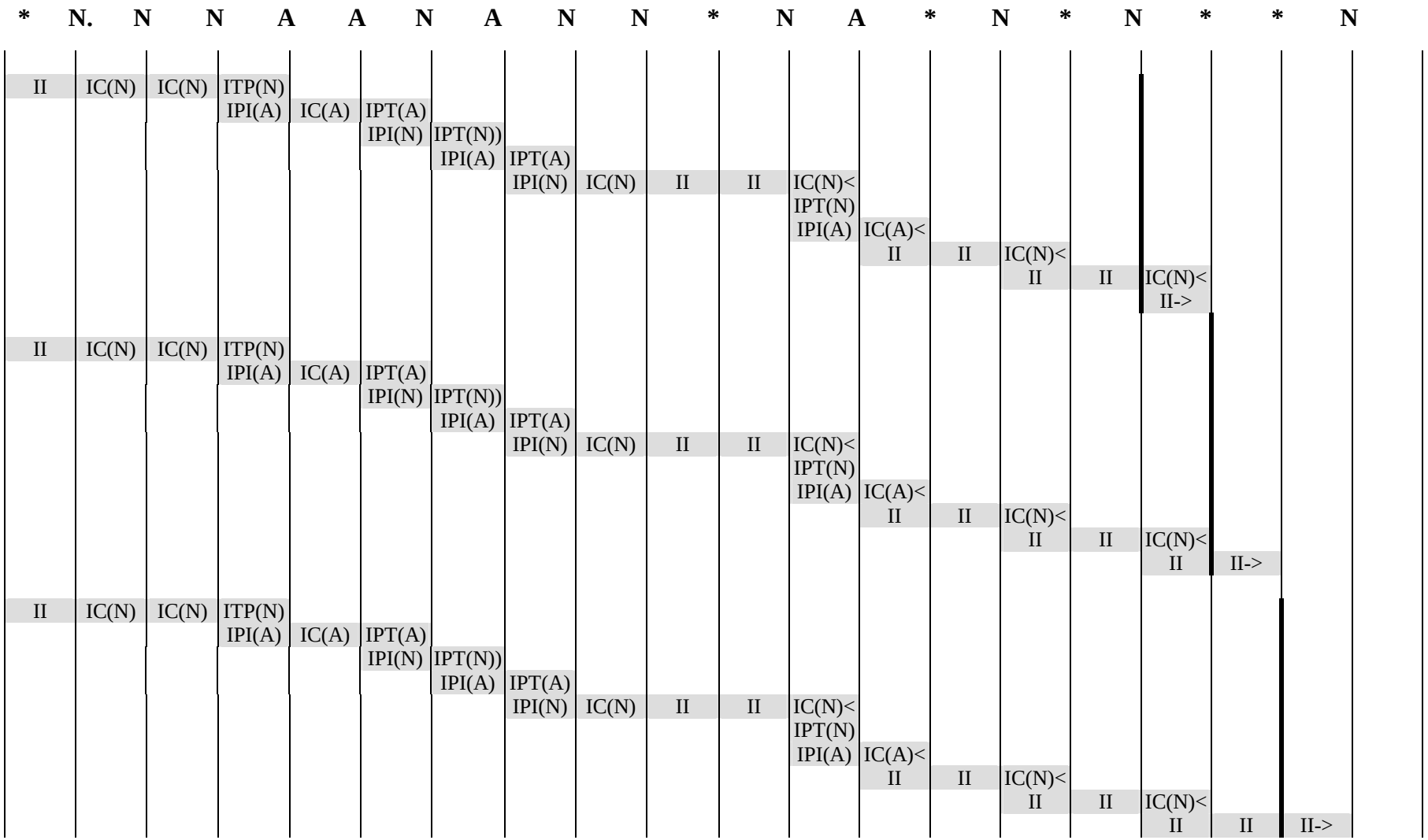
O estado N representa o estado NORMAL, A representa o estado ANOMALO e * representa o desconhecimento do estado.

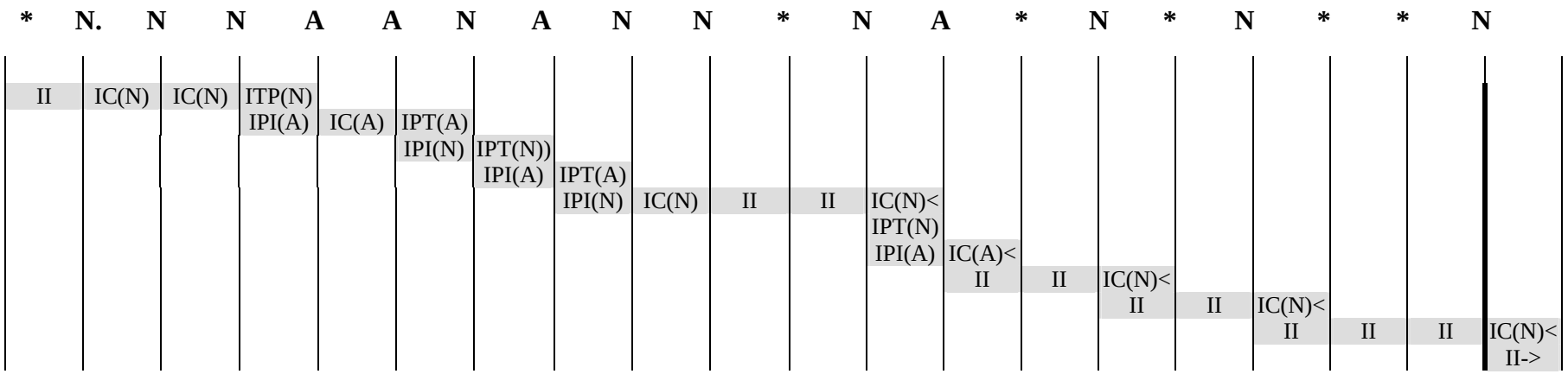
Na figura a notação “IC(X)**” significa um intervalo de certeza pontual localizado (na figura) no marco de instante a esquerda de sua localização e a notação “->” significa que o intervalo final é o tempo corrente (TC).


* N N N A A N A N N * N A * N * N * * N
II IC(N)< II->
II IC(N) II->
II IC(N) IC(N) II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IC(N)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IC(A)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N)< II->


* N. N N A A N A N N * N A * N * N * * N
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II->


* N. N N A A N A N N * N A * N * N * * N
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II II->


* N. N N A A N A N N * N A * N * N * * N
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II II IC(N)< II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II II IC(N)< II II->
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II II IC(N)< II II II->


* N. N N A A N A N N * N A * N * N * * N
II IC(N) IC(N) ITP(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(N)) IPI(A) IPT(A) IPI(N) IC(N) II II IC(N)< IPT(N) IPI(A) IC(A)< II II IC(N)< II II IC(N)< II II IC(N)< II->

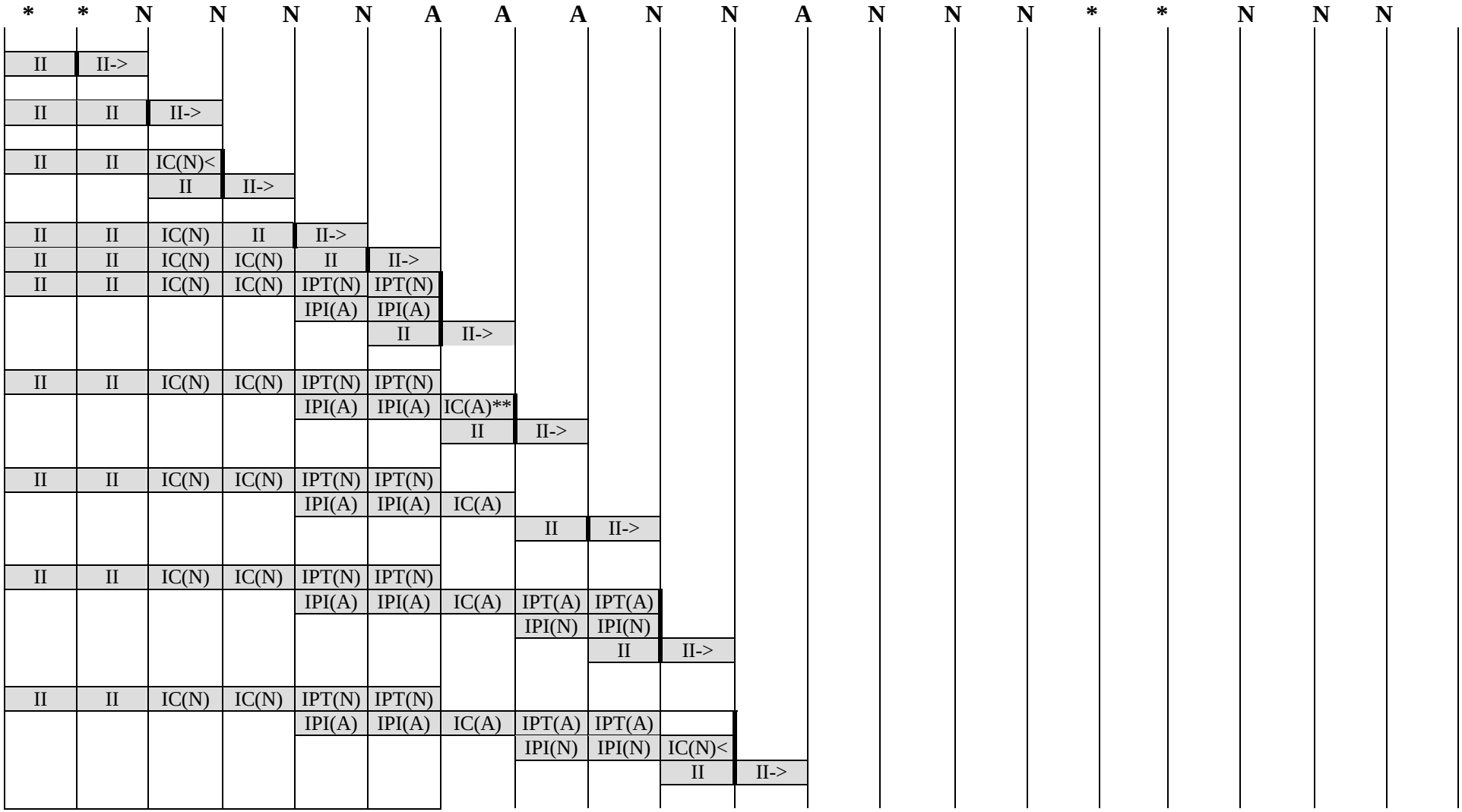
2 Exemplo #2 – Defasada em até 2 ciclos

A figura a seguir mostra um exemplo da evolução do modelamento em intervalos de tempo a partir de uma seqüência de estados observados de objetos intermediários defasados em até 2 ciclos. A seqüência de estados simulada é:

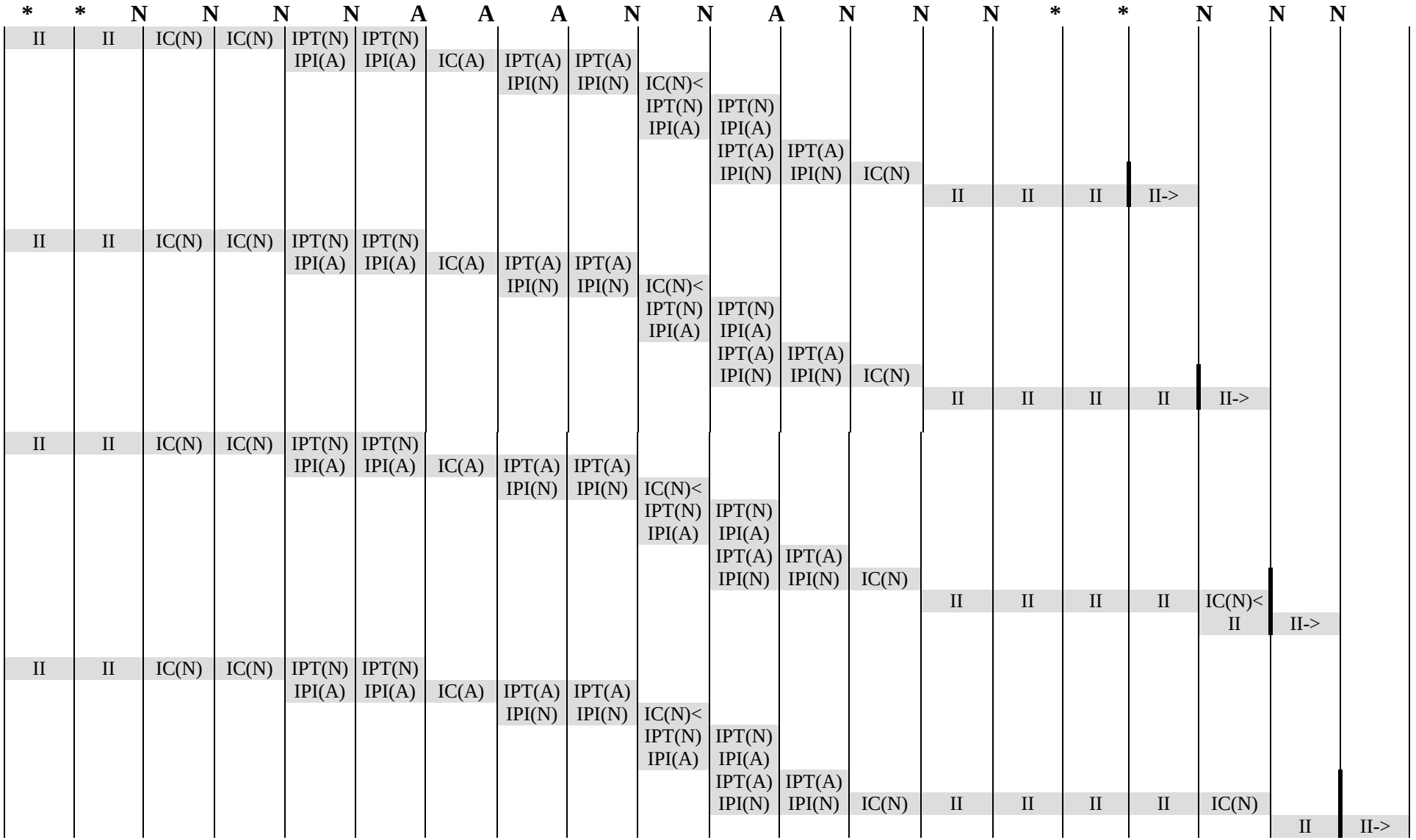
```
Seqüência de estados:
    *, *, N, N, N, N, A, A, A, N, N, A, N, N, N, *, *, N, N, N
```

O estado N representa o estado NORMAL, A representa o estado ANOMALO e * representa o desconhecimento do estado.

Na figura a notação “IC(X)**” significa um intervalo de certeza pontual localizado (na figura) no marco de instante a esquerda de sua localização e a notação “->” significa que o intervalo final é o tempo corrente (TC).


* * N N N N A A A N N A N N N * * N N N
II II-> II II-> II II IC(N)< II II-> II II IC(N) II II-> II II IC(N) IC(N) II II-> II II IC(N) IC(N) IPT(N) IPT(N) IPI(A) IPI(A) II II-> II II IC(N) IC(N) IPT(N) IPT(N) IPI(A) IPI(A) IC(A)** II II-> II II IC(N) IC(N) IPT(N) IPT(N) IPI(A) IPI(A) IC(A) II II-> II II IC(N) IC(N) IPT(N) IPT(N) IPI(A) IPI(A) IC(A) IPT(A) IPT(A) IPI(N) IPI(N) II II-> II II IC(N) IC(N) IPT(N) IPT(N) IPI(A) IPI(A) IC(A) IPT(A) IPT(A) IPI(N) IPI(N) IC(N)< II II->

*	*	N	N	N	N	A	A	A	N	N	A	N	N	N	*	*	N	N	N
II	II	IC(N)	IC(N)	IPT(N)	IPT(N)														
				IPI(A)	IPI(A)	IC(A)	IPT(A)	IPT(A)											
							IPI(N)	IPI(N)	IC(N)<										
									IPT(N)	IPT(N)									
									IPI(A)	IPI(A)									
										II	II->								
										IPI(N)	IPI(N)								
											II	II->							
II	II	IC(N)	IC(N)	IPT(N)	IPT(N)														
				IPI(A)	IPI(A)	IC(A)	IPT(A)	IPT(A)											
							IPI(N)	IPI(N)	IC(N)<										
									IPT(N)	IPT(N)									
									IPI(A)	IPI(A)									
										IPT(A)	IPT(A)								
										IPI(N)	IPI(N)	IC(N)<							
												II	II->						
II	II	IC(N)	IC(N)	IPT(N)	IPT(N)														
				IPI(A)	IPI(A)	IC(A)	IPT(A)	IPT(A)											
							IPI(N)	IPI(N)	IC(N)<										
									IPT(N)	IPT(N)									
									IPI(A)	IPI(A)									
										IPT(A)	IPT(A)								
										IPI(N)	IPI(N)	IC(N)							
													II	II->					
II	II	IC(N)	IC(N)	IPT(N)	IPT(N)														
				IPI(A)	IPI(A)	IC(A)	IPT(A)	IPT(A)											
							IPI(N)	IPI(N)	IC(N)<										
									IPT(N)	IPT(N)									
									IPI(A)	IPI(A)									
										IPT(A)	IPT(A)								
										IPI(N)	IPI(N)	IC(N)							
													II	II	II->				


* * N N N N A A A N N A N N N * * N N N
II II IC(N) IC(N) IPT(N) IPI(A) IPT(N) IPI(A) IC(A) IPT(A) IPI(N) IPT(A) IPI(N) IC(N)< IPT(N) IPI(A) IPT(N) IPI(A) IPT(A) IPI(N) IPT(A) IPI(N) IC(N)
II II II II-> II II II II-> II II II II IC(N)< II II-> II II II II IC(N) II II->

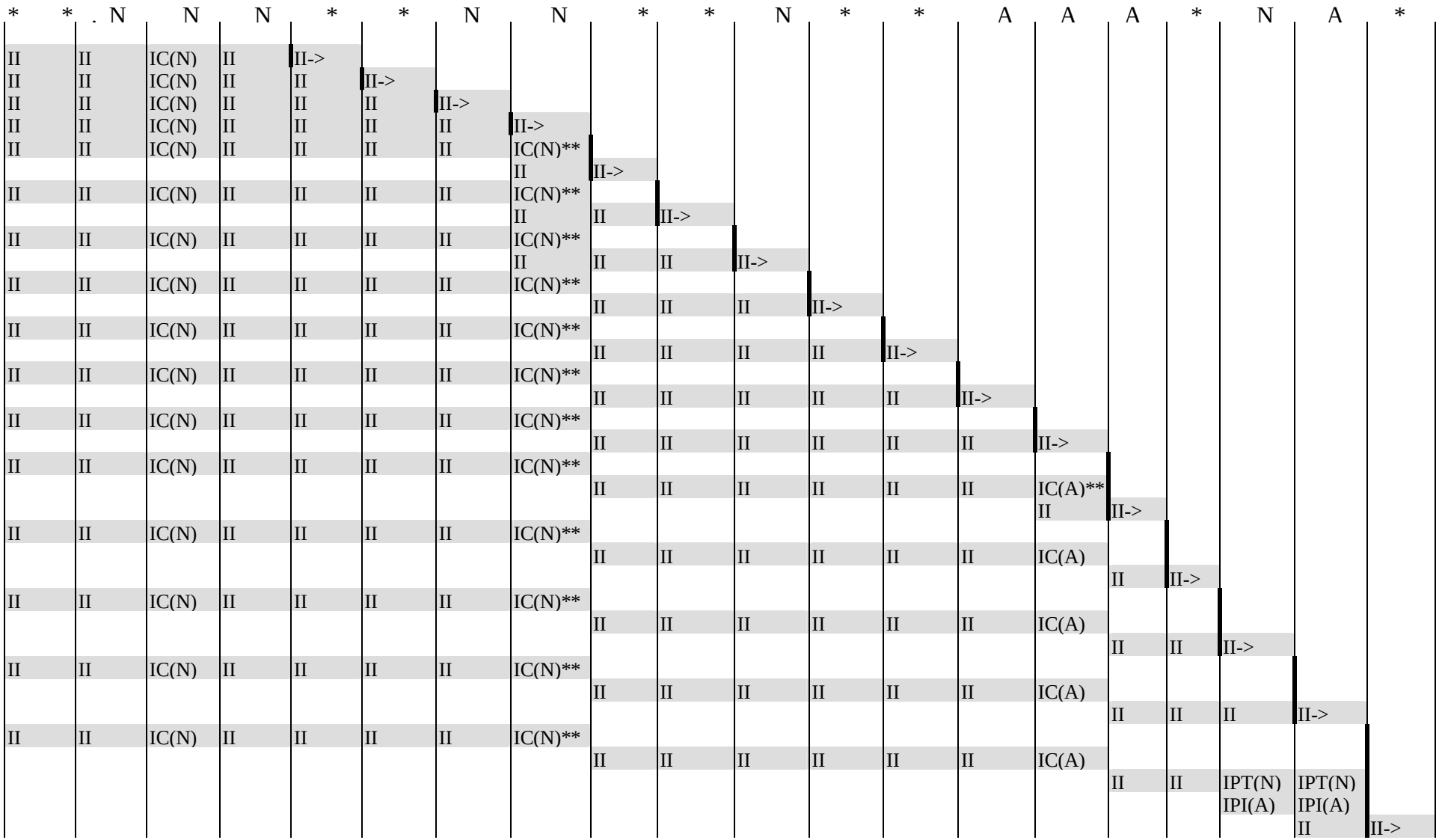
3 Exemplo #3 – Defasada em até 2 ciclos com predominância de ausência de observações

A figura a seguir mostra a simulação da evolução do modelamento em intervalos de tempo a partir de uma seqüência de estados observados de objetos intermediários defasados em até 2 ciclos.. A seqüência de estados simulada é:

```
Estados: *, *, N, N, N, *, *, N, N, *, *, N, *, *, A, A, A, *,
N, A, *
```

O estado N representa o estado NORMAL, A representa o estado ANOMALO e * representa o desconhecimento do estado.

Na figura a notação “IC(X)**” significa um intervalo de certeza pontual localizado (na figura) no marco de instante a esquerda de sua localização e a notação “->” significa que o intervalo final é o tempo corrente (TC).


* * . N N N * * N N * * N * * A A A * N A *
II II IC(N) II II-> II II IC(N) II II II-> II II IC(N) II II II II-> II II IC(N) II II II II II-> II II IC(N) II II II II IC(N)** II II-> II II IC(N) II II II II IC(N)** II II II-> II II IC(N) II II II II IC(N)** II II II II-> II II IC(N) II II II II IC(N)** II II II II II-> II II IC(N) II II II II IC(N)** II II II II II II-> II II IC(N) II II II II IC(N)** II II II II II II IC(A)** II II-> II II IC(N) II II II II IC(N)** II II II II II II IC(A) II II-> II II IC(N) II II II II IC(N)** II II II II II II IC(A) II II II-> II II IC(N) II II II II IC(N)** II II II II II II IC(A) II II IPT(N) IPT(N) IPI(A) IPI(A) II II->

Referências Bibliográficas

[ABU-HANNA 1990] ABU-HANNA, AMEEN; BENJAMINS, V. R.; Knowledge classification of models in model based diagnosis. International Workshop on Expert Systems and their Aplications, 10, General conference on second generation expert systems, **Proceedings**. Avignon. 1990. p97-110.

[ABU-HANNA 1994] ABU-HANNA, AMME; **Multiple domain models in diagnostic reasoning**. Amsterdan, 1994. 169p. Tesis (PhD), University of Amsterdam.

[ALBAGHDADI 2001] ALBAGHDADI, M.; et all.; A framework for event correlation in communication systems. In: MMNS 2001 - IFIP/IEEE International Conference on Management of Multimedia Networks and Services, 4, Chicago, IL, USA. 2001. **Proceedings**. Lecture Notes in Computer Science, v2216, Springer, 2001. p271-84 (www.mnlab.cs.depaul.edu/mmns2001/proceeding/22160271.pdf)

[ALLEN 1983] ALLEN, JAMES F. Maintaining knowledge about temporal intervals. **Communications of the ACM** v.26, n.11 p.832-43, nov 1983.

[ALLEN 1984] ALLEN, JAMES F. Towards a general theory of action and time. **Artificial Inteligence**. v.23, p.123-54, 1984.

[BARBOSA 2002] BARBOSA, J. C. F.; NOGUEIRA, J. M. S. Detecção e análise de falhas usando redes de dependências em sistemas distribuídos de gerenciamento. In: SBRC'2002 - Simpósio Brasileiro de Redes de Computadores, 20, Buzios, Rio de Janeiro, Brazil, May 2002. **Proceedings**. Rio de Janeiro, May 2002. p69-84.

[BARROS 1999] BARROS, LELIANE N.; LEMOS, MARILZA; BERNAL; VOLNYS B.; WAINER, JACQUES. Model Based Diagnosis for Network Communication Faults**.** In: AIDIN'99 - International Workshop on Artificial Intelligence for Distributed Information Networking, 3, Orlando, EUA, Jul.

1999. **Technical Report WS-99-03**, Menlo Park, California, AAAI Press, 1999, p57-62.(www.lsi.usp.br/~volnys/papers/AIDIN-99.pdf)

[BECKER 1995] BECKER, J. D.; et al. BEOWULF: A parallel workstation for scientific computation. **Proceedings**. International Conference on Parallel Processing, 1995.

[BENJAMINS 1993] BENJAMINS, RICHARD. **Problem solving methods for diagnosis**. PhD Tesis. University of Amsterdam, Amsterdam, Dec. 1993. 172p.

[BERNAL 1999a] BERNAL, VOLNYS B.; KOFUJI, SÉRGIO T.; SIPAHI, GUILHERME M.; ANDERSON, ALAN G.; PAD Cluster: an open, modular and low cost high performance computing system. In: SBAC'99 - Symposium on Computer Architecture and High Performance Computing, 11, Natal, RN, Brazil, Sep. 1999. **Procedings**. Porto Alegre, 1999, p215-22. (www.lsi.usp.br/~volnys/papers/SBAC-99-slides.pdf)

[BERNAL 1999b] BERNAL, VOLNYS B.; BARROS, LELIANE N. DE; LEMOS, MARILZA; WAINER, JACQUES; Fault diagnosis for local area network environments. In: IEEE LANOMS'99 - Latin American Network Operations and Management Symposium, 1, Rio de Janeiro, RJ, Brazil, Dec. 1999. **Proceedings**. Rio de Janeiro, 1999. p141-52 (www.lsi.usp.br/~volnys/papers/LANOMS-99.pdf)

[BERNAL 1999c] BERNAL, VOLNYS B. Implementacao de sistemas TMN e suas relacoes com a arquitetura CORBA: estado da arte e perspectivas. Relatório técnico. São Paulo, 1999. (www.lsi.usp.br/~volnys/papers/TMN-CORBA.pdf)

[BOARDMAN 2002] BOARDMAN, BRUEC. **A mom with Smarts**. Network Computing Magazine. May, 8, 2002. (http://img.cmpnet.com/nc/1316/graphics/1316f2_file.pdf)

[BULL 1997] BULL. Integrated system management: introduction to ISM architecture. **Tutorial.** 1997.

[BULL 1998] BULL. Integrated system management: introduction to management concepts. **Tutorial.** 1998.

[BRISA 1993] BRISA - Sociedade Brasileira para Interconexão de Sistemas Abertos. **Gerenciamento de redes**: uma abordagem de sistemas abertos.Makron Books, 1993.

[CISCO 1999] Cisco Inc. **Cisco network monitoring and event correlation guidelines**. 1999. (http://www.cisco.com/warp/public/cc/pd/wr2k/tech/cnm_rg.htm)

[CONSOLE 1990] Integrating models of the correct behavior into abdutive diagnosis. ECAI-90, Stockholm, 1990. **Proceedings**. ECCAI, Pitman Publishing, London, 1990.

[CONSOLE 1998a] BUSONI, V.; CONSOLE, L.; TERENZIANI, P.; DUPRË, D. A spectrum of definitions for temporal model-based diagnosis. Artificial Intelligence 102(1). 1998. **Proceedings**. p39-79. (www.di.unito.it/~dtd/papers)

[CONSOLE 1998b] CONSOLE, L.; DUPRÉ, D. THESEIDER. On the dimensions of temporal model-based diagnosis. In: 9th International Workshop on Principles of Diagnosis (Dx98), 9, Cape Cod, MA, EUA, May 1998. **Proceedings**. p16-23, (www.di.unito.it/~dtd/papers)

[DAVIS 1984] DAVIS, R. Diagnostic reasoning based on structure and behavior. **Artifitial Intelligence**, 24(1), 1984. p247-310.

[DAVIS 1988] DAVIS, RANDALL; HAMSCHER, WALTER; Model-based reasoning: Troubleshooting. In: **Exploring Artificial Intelligence**, chapter 8, p297-346. Morgan Kaufmann Publishers, 1988.

[ETHERIDGE 1998] ETHERIDGE, JASON; et. al. Distributed architecture for coss-domain network management. **Proceedings**. In: IEEE/IFIP 1998 Network Operations and Management Symposium, New Orleans, USA, 1998. p610-8.

[FERREIRA 1986] FERREIRA, AURÉLIO. Novo dicionário Aurélio da língua portuguesa. 2ª edição, Nova Fronteira, 1986. 1838p.

[FONTANINI 2002] FONTANINI, S.; WAINER, J.; BERNAL, V.; MARANGON, S.; Model based diagnosis in LANs. In: IPOM 2002 - IEEE Workshop on IP Operations and Management, Workshop, Dallas , Texas, EUA. Oct. 2002.

Proceedings. IEEE, Dallas, 2002. p221-5. (www.lsi.usp.br/~volnys/papers/IPOM-2002.pdf)

[FULTON 1998] FULTON, ROGER; DITYA, V.; JAIN, A. Simplifying the development of network management systems in a distributed environment. **Proceedings**. In: IEEE/IFIP 1998 Network Operations and Management Symposium, New Orleans, USA, 1998. p549-58.

[GAMPER 1996] GAMPER, JOHANN; **A temporal reasoning and abstraction framework for model-based diagnosis systems**. PhD. Tesis. Faculty of Mathematics, Computer Sciences and Natural Sciences, Rheinisch-Westfälischen Technischen Hochschule Aachen University, 1996. (http://www.kbs.uni-hannover.de/Arbeiten/Dissertationen/96/ gam96_phd.html).

[GAY 1995] GAY, VALÉRIE; LEYDEKKERS, P.; VELD, R. **Specification of multiparty audio and video interaction based on the reference model of open distributed processing**. Computer Networks and ISDN systems, 1995.

[GHETIE 1998] GHETIE, JOSEPH. Managed agents and agent development tools analysis and evaluation. Tutorial. In: **IEEE/IFIP 1998 Network Operations and Management Symposium**. New Orleans, Feb 1998.

[HOWARD 2001] HOWARD, LARRY; An algorithm for diagnostic reasoning using TFPG models in embedded real-time applications. In: Autotestcon 2001, Valley Forge, Pensylvania, EUA. Aug. 2001. **Proceedings**. 2001. p978-87. (www.isis.vanderbilt.edu/publications/archive/Howard_LP_8_20_2001_An_Algorit.pdf)

[JACKSON 1999] JACKSON, PETER; **Introduction to expert Systems**. Third Edition. Addison Wesley, 1999. 541p.

[JAKOBSON 1999] JAKOBSON, GABRIEL. New global solutions of event correlation based on distributed infrastructure. In: IEEE Latin American Network Operations and Management Symposium. Rio de Janeiro, RJ, Brazil. Dec. 1999. **Tutorial**. Rio de Janeiro, Brazil, 1999.

[JAKOBSON 1993] JAKOBSON, G.; WEISSMAN, M. Alarm correlation. **IEEE Network**. Nov. 1993. p52-59.

[KLEIN 1987] KEIN, DAVID; FININ, TIM; What's in a deep model? A characterization of knowledge depth in intelligent safety systems. In: IJCAI 1987. **Proceedings**. p 559-62.

[KLIGER 1995] KLIGER, S; YEMINI, S.; YEMINI, Y.; OHSIE, D. STOLFO. A Coding Approach to Event Correlation. In: International Symposium on Integrated Network Management, 4, Santa Barbara, California, EUA, 1995. **Proccedings**. (http://www.cs.columbia.edu/ids/research/keypapers/papers/eventcorrelation/isinm95.pdf)

[LEMOS 1997] LEMOS, MAILZA. **Engenharia do conhecimento aplicada ao domínio de gerenciamento de falhas em redes de comunicação: uma abordagem baseada em modelos.** Dissertação de mestrado pela Escola Politécnica da Universidade de São Paulo. São Paulo, 1997. 136p.

[LEMOS 1998] LEMOS, M. Um método de resolução de problema reusável para diagnóstico automático do domínio de gerenciamento de falhas em redes de comunicação. In: SBRC'98, Simpósio Brasileiro de Redes de Computadores, 16. **Proceedings**. p106-21, 1998.

[LEMOS 1999] LEMOS, MARILZA; BARROS, LELIANE; BERNAL, VOLNYS; WAINER, JACQUES. **Building reusable knowledge models for the communication network domain**. In: AKAW'99 - Fourth Australian Knowledge Acquisition Workshop, 4, Sidney, Australia, Cec. 1999.**Proceedings**. Sidney, Australia, 1999. (www.lsi.usp.br/~volnys/papers/AKAW-99.pdf)

[LEVI 1990] LEVI, SHEM-TAVI; AGRAWALE, ASHAK K. **Real time system design**. Singapura. McGrall-Hill, 1990 p.299.

[LIU 99] LIU, G.; MOK, A. K.; YANG, E. J. Composite Events for Network Event Correlation. **Proceedings**. In: IM'99, May 24-28, Boston, EUA. 1999. (http://www.cs.utexas.edu/users/liugt/publications/im99.html)

[MANSOURI 1995] Mansouri-Samani, M., Sloman, M. (1995) GEM A Generalised Event Monitoring Language For Distributed Systems, Research Report DoC 95/8 Imperial College. 9, 26pp (www.doc.ic.ac.uk/deptechrep/DTR95-8.pdf)

[MAZUMDAR 1998] MAZUMDAR, SUBRATA. Inter-domain management: CORBA, OSI, SNMP. Tutorial. In**: IEEE/IFIP 1998 Network Operations and Management Symposium**. New Orleans, Feb 1998

[MEIRA 1997a] MEIRA, DILMAR M. **A model for alarm correlation in telecommunications networks**. Phd Thesis. Department of Computer Science. Federal University of Minas Gerais (UFMG). Belo Horizonte, Nov. 1997. 149p. (http://www.sis.dcc.ufmg.br/sis2/apresentacoes.html)

[MEIRA 1997b] MEIRA, DILMAR M.; NOGUEIRA, J. M. S. Métodos e algoritmos para correlação de alarmes em redes de telecomunicações. In: Simpósio Brasileiro de Redes de Computadores, 15, São Carlos, 1997. **Proceedings**. São Carlos, 1997. p79-89. (http://www.sis.dcc.ufmg.br/sis2/apresentacoes.html)

[MOGHÉ 1998] MOGHÉ, P.; EVANGELISTA, M.; RAP: rate adaptative polling for network management applications. In: NOMS 1998 – IEEE/IFIP Network Operations and Management Symposium. New Orleans. LA. EUA. Feb. 1998. **Proceedings**. v2. IEEE. 1998.

[OLIVEIRA 1998] OLIVEIRA, JONAS SANTIAGO. **Análise das restrições para gerenciamento de servidores em redes WAN de baixa velocidade.** Dissertação de mestrado. Escola Politécnica da USP. 1998.

[OMG 1998] OMG. **JIDM Interaction Translation**. OMG Document Number: telecom/98-10-10. Object Management Group - Telecom Task Force. 206p. Oct, 1998. (from http://www.omg.org).

[OMG 1996] OMG. **Corba-based telecommunication network management system**. OMG white paper. Object Management Group - Telecom Task Force. 29p. May, 1996. (from http://www.omg.org).

[OHSIE 1997a] OHSIE, D.; MAYER, A.; KLINGER, S.; YEMINI, S. Event modeling with the MODEL language. In: IFIP IEEE International Symposium

on Integrated Network Management, 5, San Diego, CA, USA. 1997, **Proceedings**. p625-37.

[OHSIE 1997b] OHSIE, D.; MAYER, A.; KLINGER, S.; YEMINI, S. **Event modeling with the MODEL language**: an tutorial introduction, 1997, 14p (www.cs.columbia.edu/ids/research/keypapers/papers/eventcorrelation/inm97cr5.pdf)

[RESENDE 1996] RESENDE, A.; WAINER, J. A temporal extension to the parsimonious covering theory. SBIA'96, Brazilian Conference on Artifitial Inteligence, 13. **Proceedings**. Volume 1159. Lecture notes on AI, p201-210, Springer-Verlag, 1996. (http://www.ic.unicamp.br/~wainer/papers/sbia96b.ps.gz)

[RFC1155 1990] RFC 1155. **SMI - Structure and Identification of Management Information for TCP/IP based Internets**, 1990. (http://www.ietf.org/rfc/rfc1155.txt)

[RFC1157 1990] RFC 1157. **SNMP - A Simple Network Management Protocol**, 1990. (http://www.ietf.org/rfc/rfc1157.txt)

[RFC1212 1991] RFC 1212. **Concise MIB Definitions**, 1991. (http://www.ietf.org/rfc/rfc1212.txt)

[RFC1213 1991] RFC 1213. **Management Information Base for Network Management of TCP/IP-based Internets. (MIBII)**. 1991. (http://www.ietf.org/rfc/rfc1213.txt)

[RFC1214 1991] RFC 1214. **OSI Internet Management: MIB**, 1991. (http://www.ietf.org/rfc/rfc1214.txt)

[RFC1215 1991] RFC 1215. **A Convention for defining traps for use whith the SNMP**, 1991. (http://www.ietf.org/rfc/rfc1215.txt)

[RFC1757 1995] RFC 1757. **Remote network monitoring management information base.** Feb. 1995. (http://www.ietf.org/rfc/rfc1757.txt)

[RFC1901 1996] RFC 1901. **Introduction to Community-Base SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1901.txt)

[RFC1902 1996] RFC 1902. **Structure of Management Information for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1902.txt)

[RFC1903 1996] RFC 1903. **Textual Conventions for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1903.txt)

[RFC1904 1996] RFC 1904. **Conformance Statements for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1904.txt)

[RFC1905 1996] RFC 1905. **Protocol Operations for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1905.txt)

[RFC1906 1996] RFC 1906. **Transport Mappings for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1906.txt)

[RFC1907 1996] RFC 1907. **Management Information Base for SNMPv2**. Jan. 1996. (http://www.ietf.org/rfc/rfc1907.txt)

[RFC1908 1996] RFC 1908. **Coexistence Between Version 1 and Verson 2 of the Internet-Standard Network Management Framework**. Jan. 1996. (http://www.ietf.org/rfc/rfc1908.txt)

[RFC2271 1998] RFC 2271. **An Architecture for Describing SNMP Management Frameworks,** Jan. 1998. (http://www.ietf.org/rfc/rfc2271.txt)

[RFC2272 1998] RFC 2272. **Message Processing and Dispatching for the simple network management protocol (SNMP)**, Jan. 1998. (http://www.ietf.org/rfc/rfc2272.txt)

[RFC2273 1998] RFC 2273. **SNMPv3 applications**, Jan. 1998. (http://www.ietf.org/rfc/rfc2273.txt)

[RFC2274 1998] RFC 2274. **User-based security model (USM) for version 3 of the simple network management protocol**, Jan. 1998. (http://www.ietf.org/rfc/rfc2274.txt)

[RFC2275 1998] RFC 2275. **View-based access control model (VACM) for the simple network management protocol,** Jan. 1998. (http://www.ietf.org/rfc/rfc2275.txt)

[RFC2021 1997] RFC 2021. **Remote netowork monitoring management information base:** version 2 using SMIv2. Jan. 1997. (http://www.ietf.org/rfc/rfc2021.txt)

[RFC2593 1999] RFC 2593. **Script MIB extensibility protocol version 1.0**. May. 1999. (http://www.ietf.org/rfc/rfc2593)

[RFC2613 1999] RFC 2613. **Remote network monitoring MIB extensions for switchd networks**: version 1.0. Jun. 1999. (http://www.ietf.org/rfc/rfc2613.txt)

[RFC2925 2000] RFC 2925. **Definitions of managed objects for remote ping, traceroute, and lookup operations**. Sep. 2000. (http://www.ietf.org/rfc/rfc2925)

[RFC2981 2000] RFC 2981. **Event MIB**. Oct 2000. (http://www.ietf.org/rfc/rfc2981)

[RFC2982 2000] RFC 2982. **Distributed Management Expression MIB**. Oct. 2000. (http://www.ietf.org/rfc/rfc2982)

[RFC3014 2000] RFC 3014. **Notification log MIB**. Nov. 2000. (http://www.ietf.org/rfc/rfc3014)

[RFC3165 2000] RFC 3165. **Definitions of managed objects for the delegation of management scripts**. Aug. 2001. (http://www.ietf.org/rfc/rfc3165)

[RFC3231 2002] RFC 3231. **Definitions of managed objects for scheduling management operations**. Jan. 2002. (http://www.ietf.org/rfc/rfc3231)

[ROSE 1996] ROSE, MARSHALL T. **The simple book:** an introduction to network management. Revised second edition. Prentice Hall 1996. p289.

[SAYDAM 1998] SAYDAM, TUNCAY. Service management value-added services and business management. Tutorial. In: **IEEE/IFIP 1998 Network Operations and Management Symposium**. New Orleans, Feb 1998

[SMARTS 2000] System Management Arts (SMARTS). **Root cause analysis and its role in event management**. White paper. Cisco Magazine. Sep. 2000. (http://www.ciscoworldmagazine.com/webpapers/2000/09_smarts.shtml)

[SORTICA 1999] SORTICA, EDUARDO. **Redes de Telecomunicaçòes TMN e Gerência Integrada de Redes e Serviços**. 265p. 1999.

[STALLINGS 1996] STALLINGS, W. **SNMP, SNMPv2 and RMON**: Pratical network management. Second edition. Addison-Wesley 1996.

[WAINER 2000] WAINER, JACQUES; BARROS, L.; BERNAL, V. **Network fault diagnosis: a model based approach.** In: NOMS'2000 - Latin American Network Operations and Management Symposium, 11, Haway, EUA, Apr. 2000. **Poster**. 2000.(www.lsi.usp.br/~volnys/papers/NOMS-2000.pdf)

[WHITE 1998] WHITE, TONY. **Alarm Correlation**. Carleton University, Systems Engineering. Tutorial. 1998. (http://www.sce.carleton.ca/ftp/pub/94588/alarmcor.ppt)

[WIELEMAKER 2002] WIELEMAKER, JAN. **SWI-Prolog 5.0: Reference Manual**. 2002. (www.swi.psy.uva.nl/cgi-bin//nph-download/SWI-Prolog/refman/refman.pdf)

[XOPEN] X/OPEN. **Inter-domain Management: Specification Translation**.

[ZHENG 2002] ZHENG, OINGGUO, XU, KE; LV, WEINGENG; MA, SHILONG. Inteligent Search of Correlated Alarms for GSM Networks with Model-based Constraints. 9th IEEE International Conference on Telecommunications 2002. **Proceedings**. (http://www.nlsde.buaa.edu.cn/~kexu/papers/ict02.pdf)