

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CLAIRMONT BORGES

Serviços para Auxiliar Decisão mediante Incerteza

Tese apresentada como requisito parcial para a
obtenção do grau de Doutor em Ciência da
Computação

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, março de 2005.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Borges, Clairmont

Serviços para Auxiliar Decisão mediante Incerteza / Clairmont Borges - Porto Alegre: Programa de Pós-Graduação em Computação, 2005.

131f.:il.

Tese (doutorado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Ciência da Computação. Porto Alegre, BR-RS, 2005. Orientador: Cláudio Fernando Resin Geyer.

1. Representação de Conhecimento Incerto. 2. Decisão Mediante Incerteza. 3. Aprendizagem por Reforço. 4. Probabilidade. 5. Serviços Incrementais. I. Geyer, Cláudio Fernando Resin. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-reitor: Prof. Pedro Cezar Dutra da Fonseca

Pró-Reitora Adjunta de Pós-Graduação: Profa. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Phillipe Olivier Alexandre Navaux

Coordenador do PGCC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a DEUS,

aos Amigos Fiéis ...

aos Verdadeiros Gaúchos – os que amam a terra que os acolhe, seja onde for ...

... através do cultivo das amizades sinceras,

do respeito aos vizinhos,

dos valores de família.

e dedico esta tese aos meus pais:

Mário e Ivonete.

SUMÁRIO

LISTA DE FIGURAS	6
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Motivação	12
1.1.1 Cenários de Consumo de Informação.....	13
1.2 Uma Breve Revisão de Algumas Abordagens Anteriores	14
1.2.1 Abordagens Numéricas	15
1.2.2 Abordagens Simbólicas.....	15
1.2.3 Observações sobre as Abordagens	16
1.2.4 As Principais Limitações das Abordagens Anteriores	17
1.3 O Conteúdo desta Proposta de Tese	17
2 CONCEITOS BÁSICOS E ESTADO DA ARTE	19
2.1 Introdução à Probabilidade	19
2.2 Variável Aleatória e Distribuição de Probabilidades	21
2.3 Probabilidade Condicional e Regra de Bayes	25
2.4 Regra da Cadeia, Diagrama de Árvore e Teorema de Bayes	27
2.5 Redes Probabilísticas	31
2.6 Tipos Básicos de Inferência Probabilística	37
2.6.1 Inferência Causal (ou <i>top-down</i>)	38
2.6.2 Inferência de Diagnóstico (ou <i>bottom-up</i>).....	40
2.6.3 Inferência Intercausal (ou Explaining Away)	42
2.7 Analogia da Rede Probabilística com o Modelo Relacional	43
2.7.1 Dificuldade do Modelo Relacional para Controlar Semântica	47
2.7.2 Dificuldade da Rede Probabilística para Controlar Semântica	48
2.8 Conclusões	49
3 SERVIÇOS PARA AUXILIAR DECISÃO MEDIANTE INCERTEZA	50
3.1 Descrição do Problema	50
3.2 Solução do Problema	51
3.2.1 O <i>Framework</i> SAIC	51
3.2.2 <i>Framework</i> SADI – Arquitetura de Serviços	55
3.2.3 O <i>Framework</i> SADI – Representação de Conhecimento	77

3.2.4	Computação Contextual e Metodologia Contextual.....	86
4	ESTUDO DE CASO: AGENTE DE SOFTWARE	87
4.1	O Objetivo do Agente de Software	88
4.2	Metadados: Representação e Intercâmbio	91
4.3	Serviço de Percepção do Ambiente do Agente de Software	94
4.4	Serviço de Detecção de Contexto do Agente de Software	103
4.4.1	<i>Entrada de Percepção e o Serviço de Detecção de Contexto.....</i>	103
4.4.2	Modelo de Mundo do Assistente de <i>Feedback</i>	106
4.4.3	Relacionamento entre Percepção e Modelo de Mundo	114
4.5	Serviço de Aprendizagem do Agente de Software	116
4.5.1	Rastreamento do Conteúdo.....	116
4.5.2	<i>Feedback</i> Interior Indireto e Direto	117
4.6	Serviço de Tomada de Decisão do Agente de SW.....	120
4.6.1	Conteúdo Candidato	121
5	CONCLUSÃO.....	125
5.1	Trabalhos Futuros	126
	REFERÊNCIAS	128

LISTA DE FIGURAS

Figura 2.1: Relações de Conjuntos.	21
Figura 2.2: Exemplo de Diagrama de Árvore.	28
Figura 2.3: Exemplo de Partições no Espaço Amostral.	29
Figura 2.4: Visualização do Exemplo de Processo Estocástico.	29
Figura 2.5: Exemplo de Atualização de Crenças (usando evidências E_i).	31
Figura 2.6: Exemplo de Rede Probabilística.	32
Figura 2.7: Rede Probabilística e sua DPC equivalente.	33
Figura 2.8: Conexão serial. Quando B é instanciado, ele bloqueia a comunicação entre A e C	34
Figura 2.9: Conexão Divergente. Se A é instanciado, ele bloqueia a comunicação entre seus “filhos”.	34
Figura 2.10: A variável <i>Sexo</i> influencia as variáveis <i>Comp. Cabelo</i> e <i>Estatura</i>	35
Figura 2.11: Conexão Convergente. Se A for evidência, ela abre a comunicação entre seus “pais”.	35
Figura 2.12: <i>Infeção (Salmonella)</i> e <i>Gripe</i> podem causar <i>Náusea</i> , que por sua vez causa <i>Palidez</i>	36
Figura 2.13: Os Três Casos de d-Separação numa Rede Probabilística.	37
Figura 2.14: Exemplos dos tipos básicos de Inferência Probabilística.	37
Figura 2.15: Diagrama de Árvore do Exemplo de Inferência Causal.	39
Figura 2.16: Diagrama de Árvore do Cálculo da Probabilidade Marginal.	41
Figura 2.17: A) Exemplo de Rede Probabilística. B) Modelo de Entidades e Relacionamentos e Respectivas Tabelas.	45
Figura 2.18: Exemplo 1: A) Diagrama da Inferência. B) Relacionamento das Tabelas.	46
Figura 2.19: Exemplo 2: A) Diagrama da Inferência. B) Relacionamento das Tabelas.	47
Figura 2.20: Operações de Marginalização.	48
Figura 3.1: O Serviço de Busca.	52
Figura 3.2: O Serviço de Filtragem.	53
Figura 3.3: a) Operação de Pesquisa; b) Operação de Filtragem.	54
Figura 3.4: Transmissões “Vertical” e “Horizontal”.	55
Figura 3.5: A) <i>Símbolos</i> numa sub-árvore de percepção; B) <i>Espaço de Percepção</i>	58
Figura 3.6: Exemplo de contagem de <i>Tempo Contextual</i>	62
Figura 3.7: A) Serviço de Filtragem SAIC B) Serviço de Detecção de Contexto SADI C) A Propagação de Símbolos Compostos.	64
Figura 3.8: A) Serviço de Filtragem SAIC B) Serviço de Detecção de Contexto SADI.	65

Figura 3.9: A) Exemplo de Modelo de Mundo; B) Fluxo de Percepção e relacionamentos entre o Modelo de Mundo e o Espaço de Percepção.....	68
Figura 3.10: Exemplo dos subespaços <i>PO,PA,PI</i>	68
Figura 3.11: Exemplo de uma instância de classe <i>CCEL</i>	71
Figura 3.12: Exemplo de Decisão Incremental.	73
Figura 3.13: Fluxos de Feedforward e Feedback no SADI.	76
Figura 3.14: Probabilidades e Probabilidades Condicionais.	79
Figura 3.15: Probabilidades Condicionais Representadas e Não-representadas.	80
Figura 3.16: Visão Multidimensional da <i>Representação de Conhecimento SADI</i>	81
Figura 3.17: Visão Multidimensional sob o <i>Relacionamento-Transição</i>	82
Figura 3.18: Visão Multidimensional sob a Perspectiva de Causa-e-Efeito.	83
Figura 3.19: A) Exemplo de Rede Bayesiana; B) Exemplo de Árvore de Junção que resultou desta Rede Bayesiana; C) Visualização das relações de conjunto resultadas da fatorização da DPC. D) Relações de conjunto na <i>Representação de Conhecimento SADI</i>	85
Figura 4.1: Contextos de Exclusão e Inclusão.....	90
Figura 4.2: Padrão <i>Chanel Definition Format</i> para Conteúdo Filtrado.....	90
Figura 4.3: A Base dos Metadados e Suas Aplicações.....	92
Figura 4.4: Visão dos Conjuntos de Metadados.	92
Figura 4.5: Influência das Ontologias.	93
Figura 4.6: A Interface do <i>DOM</i>	94
Figura 4.7: Ontologias na Comunicação entre Agentes de Software.....	94
Figura 4.8: Relacionamento entre os <i>Contextos</i> : Localidades, Funcionalidades e Sugestão (Exclusão e Inclusão).	98
Figura 4.9: Particionamento do <i>Espaço de Contexto</i> multidimensional do Assistente.....	99
Figura 4.10: Exemplo de <i>Símbolo Composto</i> no Assistente de <i>Feedback</i>	99
Figura 4.11: Exemplo de Distribuição de Evento no Assistente de <i>Feedback</i>	100
Figura 4.12: Eventos Percebidos no Editor e Navegador <i>Web</i>	100
Figura 4.13: Eventos Percebidos no Correio, Agenda e Gerenciador de Arquivos.	101
Figura 4.14: Representação dos objetos percebidos pelo Assistente de <i>Feedback</i>	101
Figura 4.15: Relação dos atributos de objetos percebidos com as Classes <i>CCEL</i>	102
Figura 4.16: Coleta de eventos no Ambiente do Assistente de <i>Feedback</i>	102
Figura 4.17: Organização do Subespaço de Percepção de Ações (<i>PA</i>).	104
Figura 4.18: Construção dos subespaços <i>PO, PA e PI</i> do Assistente de <i>Feedback</i>	105
Figura 4.19: Organização dos cubos da estrutura multidimensional.....	105
Figura 4.20: Exemplo de Modelo de Comportamento do Usuário.	107
Figura 4.21: Exemplo dos Níveis do Modelo de Comportamento do Usuário.	107
Figura 4.22: Exemplo de Modelo de Comportamento no Navegador <i>Web</i>	109
Figura 4.23: Exemplo de Modelo de Comportamento no Correio Eletrônico.	109
Figura 4.24: Exemplo de Modelo de Comportamento no Gerenciador de Arquivos.	110
Figura 4.25: Exemplo de Modelo de Comportamento no Editor de Textos.	110
Figura 4.26: Exemplo de Modelo de Comportamento na Agenda.	111
Figura 4.27: Processamento de Percepções Incertas no Assistente de <i>Feedback</i>	112
Figura 4.28: 1.o Nível do Modelo de Mundo do Assistente de <i>Feedback</i>	113
Figura 4.29: Níveis Seguintes do Modelo de Mundo do Assistente de <i>Feedback</i>	113
Figura 4.30: Subestados de ação embutidos no estado 'Modela_Comportamento'.....	114

Figura 4.31: Rastreamento de Padrão de Comportamento Relacionado com o Termo T	118
Figura 4.32: Indexação do Termo T rastreado nos <i>Tempos Contextual</i> T_C e T_{C+J}	119
Figura 4.33: Exemplo de Uso da Ferramenta de Anotação.....	123

LISTA DE TABELAS

Tabela 2.1: Resultados dos testes com 3 equipamentos.....	22
Tabela 2.2: Probabilidades dos Resultados dos Testes	22
Tabela 2.3: Distribuição de probabilidades da variável aleatória X	22
Tabela 2.4: Exemplo de Tabela de Distribuição de Probabilidade Conjunta (DPC).	23
Tabela 2.5: Exemplo de Cálculo de Probabilidade Marginal.	24
Tabela 2.6: Exemplo de DPC sobre Cárie e Dor de Dente.	25
Tabela 2.7: Valores das probabilidades <i>à priori</i> e condicionais.	30
Tabela 2.8: Probabilidades condicionais (<i>à posteriori</i>) de todas as hipóteses, dada a evidência E_1	30
Tabela 2.9: Probabilidades condicionais (<i>à posteriori</i>) de todas as hipóteses, dada a evidência E_2	31

RESUMO

O objetivo deste trabalho é apresentar um modelo eficiente de representação de conhecimento que é construído a partir de relações de causa e efeito entre percepções e ações. Assume-se que é possível perceber o ambiente, é necessário fazer decisões mediante incerteza, é possível perceber a realimentação (*feedback*) referente ao sucesso ou fracasso das ações escolhidas, e é possível aprender com a experiência.

Nós descrevemos uma arquitetura que integra o processo de percepção do ambiente, detecção de contexto, tomada de decisão e aprendizagem, visando obter a sinergia necessária para lidar com as dificuldades relacionadas.

Além da descrição da arquitetura, é apresentada de forma sucinta uma metodologia chamada *Computação Contextual*, composta por duas fases principais: Definição e Operação. A fase de Definição envolve o projeto e modelagem de: i) Os subespaços de conhecimento conceitual e canônico; e ii) As regras de crescimento dinâmico. A fase de Operação complementa (isto é, estende e adapta) as definições iniciais através da aprendizagem feita pela interação com o ambiente.

Palavras-Chave: Representação de Conhecimento, Decisão mediante Incerteza, Aprendizagem por Reforço, Probabilidade Condicional, Serviços Incrementais.

Services to Aid Decision under Uncertainty

ABSTRACT

The objective of this work is to present an efficient model of knowledge representation that is constructed from the cause and effect relations between perceptions and actions. It is assumed that it is possible to perceive the environment, it is necessary to make decisions under uncertainty, it is possible to perceive the feedback concerning the success or failure of the chosen actions, and it is possible to learn from experience.

We describe an architecture that integrates the processes of environment perception, context-detection, decision-making and learning, in order to achieve the synergy necessary to deal with the related difficulties.

Besides the architecture description, it is briefly presented a methodology named *Contextual Computing*, composed of two main phases: Definition and Operation. The Definition phase involves the design and modeling of: i) The conceptual and canonical knowledge subspaces; and, ii) The dynamic growth rules. The Operation phase complements (*i.e.* extends and adapts) the initial definitions by learning through interaction with the environment.

Keywords: Knowledge Representation, Decision under Uncertainty, Reinforcement Learning, Conditional Probability, Scaleable Services.

1 INTRODUÇÃO

Este capítulo é iniciado com a introdução da motivação deste trabalho (seção 1.1), seguida por uma breve revisão de algumas técnicas e formalismos usados para modelar e operar com conhecimento incerto (seção 1.2), e depois é apresentada uma visão geral do conteúdo desta proposta de tese (seção 1.3).

1.1 Motivação

O crescimento contínuo da Internet por si mesmo prova suas viabilidades técnica e econômica, onde governos e iniciativas privadas garantem o desenvolvimento da infraestrutura de rede. A humanidade depende cada vez mais dos computadores e dos serviços de telecomunicação, e as produções artísticas, científicas e tecnológicas estão cada vez mais baseadas em formato digital. Um vez que a Internet está se tornando o meio de comunicação de massa da *verdadeira* Democracia, e sendo a comunicação tão fundamental para alcançar o consenso e a harmonia entre os diversos povos e comunidades, então pode-se concluir que é extremamente importante aprimorar os sistemas de comunicação mediados por computador.

Um exemplo de esforço nesta direção foi um trabalho do nosso grupo de pesquisa, em parceria com o governo do Estado do Rio Grande do Sul (através da empresa PROCERGS), onde são desenvolvidos Canais de Informação com Serviços de Filtragem (CISF) no software Direto [BAL 2002]. Durante este trabalho, foi possível reforçar o entendimento do quanto a soberania de um país Democrático está relacionada com o domínio das tecnologias estratégicas, mais especificamente as que suportam os serviços de telecomunicação. Sendo assim, uma parte da experiência adquirida neste trabalho foi sintetizada na análise: Estratégias de Governo para Promover o Desenvolvimento de Software Livre [BOR 2003a].

Por outro lado, ao longo destes esforços de pesquisa, experiências complementares foram sendo acumuladas, aproveitando as oportunidades únicas de atuar muito próximo das principais estruturas de poder da sociedade. Parte destas experiências adicionais envolvem os relacionamentos entre: Internet, Software Livre, Serviços Jornalísticos, Controle da Privacidade e da Segurança. Portanto, dando continuidade ao esforço de divulgação das experiências de pesquisa, uma oficina foi organizada e apresentada no Fórum Social Mundial 2003 [BOR 2003b].

De uma forma geral, os cenários de consumo de informação (mediado por computador) estão descritos na subseção 1.1.1. Todavia, existe um problema conhecido e intrínseco a estes cenários de consumo de informação, que é a dificuldade que um ser humano lida enquanto se esforça para encontrar ou selecionar a informação do seu interesse. Visando minimizar esta dificuldade, serviços de agentes de software têm sido desenvolvidos para auxiliar usuários humanos, e dentro do contexto das nossas pesquisas, foi desenvolvido um trabalho sobre um agente de software do tipo assistente pessoal [MEL 2002] para auxiliar usuários dos serviços de Canais de Informação com Serviços de Filtragem no software Direto [BAL 2002]. Contudo, independente do tipo de ações ou tarefas automatizadas que podem ser feitas por agentes de software, primeiramente é necessário modelar o comportamento dos usuários humanos para que seja possível inferir suas preferências e necessidades.

Soluções para a modelagem do comportamento de usuários tem sido desenvolvidas em diversas áreas de pesquisa, tais como: inteligência artificial,

psicologia, lingüística, interação homem-máquina e ciência da informação [HOR 97]. Estas áreas de pesquisa buscam, dentre outras coisas, viabilizar aplicações de software que se adaptam aos usuários, ajudando-os em suas necessidades através da construção, manutenção e utilização de modelos (de comportamento) de usuários. Nos nossos cenários de consumo de informação mediado por computador, estes modelos refletem o perfil de consumo de informação dos usuários.

Todavia, segundo [HOR 97], a incerteza é inerente a estes cenários de consumo de informação, principalmente nas tentativas de inferir as preferências e intenções das pessoas. A identificação de preferências e intenções de usuários durante a interação destes com sistemas de comunicação mediados por computador, usando evidências como as ações do usuário e os contextos destas ações, é um problema típico de tomada de decisão mediante incerteza. A construção de um modelo de usuário usando padrões de dados é uma instância de um processo mais geral denominado aprendizagem de máquina ou descoberta de conhecimento (onde a incerteza também é intrínseca ao processo de construção do modelo).

Ao longo da interação do usuário com um sistema de comunicação mediado por computador, são gerados dados que contém padrões sobre o comportamento dos usuários, e alguns exemplos desses dados são ilustrados nos nossos cenários de consumo de informação.

1.1.1 Cenários de Consumo de Informação

Uma parte do nosso cenário de consumo de informação é o *World Wide Web* (WWW, ou simplesmente *Web*). Como em qualquer outro sistema de hipertexto, a *Web* contém hiperdocumentos que possuem ligações (*links*) com outros hiperdocumentos. Estes hiperdocumentos, também chamados de páginas *Web*, são usualmente armazenados em servidores *Web* na Internet. O ato de acessar diferentes hiperdocumentos, através das ligações entre eles, é chamado de “navegação” (ou, *Web browsing*). A “navegação” é iniciada quando o usuário possui o endereço de alguma página *Web* (que pode ter sido indicado por outro usuário ou então anunciado em outra mídia). Durante a “navegação”, padrões do comportamento de um usuário podem ser observados nos seguintes locais:

- O arquivo de registro (*log*) de acesso das seções de “navegação”. Este *log* pode ser obtido do servidor *Web*, do histórico armazenado no programa de “navegação” (*browser*), da memória *cache* do *browser*, e/ou de alguma memória *cache* de um servidor de *proxy*. Este *log* contém, dentre outros dados, o endereço eletrônico das páginas acessadas, o endereço eletrônico de quem fez o acesso, e a data e hora do acesso.
- O conteúdo das páginas *Web* acessadas, o qual inclui *links*, dados multimídia (texto, som, imagens e vídeos) e também metadados baseados na linguagem *XML* (alguns exemplos podem ser encontrados em [BOR 99]).
- Os endereços de páginas *Web* (chamados “Favoritos” ou *bookmarks*), que o usuário registra no seu *browser* durante a navegação.

Adicionalmente, durante a “navegação”, o usuário acessa um universo de sistemas de informação que foram integrados pela *Web* (como descrito em [BOR 97]). Estes sistemas geram páginas *Web* dinamicamente, onde algumas delas contêm campos (*forms*) para os usuários submeterem dados ou consultas. Máquinas de busca (*search engines*) são exemplos representativos deste sistemas, e também estas máquinas complementam o processo de “navegação” na *Web*, provendo endereços de páginas

Web cujo conteúdo é idêntico ou muito similar ao conteúdo das consultas submetidas pelos usuários “navegadores” da *Web*. Novamente, padrões do comportamento de um usuário podem ser observados no:

- No arquivo de *log* gerado durante as sessões de consultas, o qual pode estar armazenado nos servidores *Web* que recebem as consultas, nos sistemas de informação que processam estas consultas, e também pode ser armazenado por agentes de software ou aplicações que estão ajudando os usuários a realizarem o consumo de informação. Este *log* contém, dentre outros dados, as palavras-chave das consultas, os seus operadores e parâmetros usados para combinar as palavras-chave ou reduzir o escopo das consultas, e a data e hora da submissão da consulta.

Outra parte deste cenário de consumo de informação é composto por sistemas de comunicação mediado por computador que já existiam antes da *Web*, incluindo os grupos de notícias (*newsgroups*), correio eletrônico (*email*) e servidores de listas. Assim como no caso dos endereços de páginas *Web*, os endereços eletrônicos desta vasta população de sistemas (e usuários destes sistemas) são descobertos pelo usuário através da indicação de alguém ou por anúncio na *Web* ou qualquer outra mídia. No caso dos grupos de notícias ou servidores de listas, um usuário se inscreve em um ou mais tópicos (ou assunto), para poder receber continuamente mensagens na medida em que elas são submetidas a estes tópicos. Neste contexto, padrões de comportamento de um usuário podem ser observados no conteúdo das mensagens e nas ações que o usuário realiza, por exemplo:

- O usuário apaga uma mensagem imediatamente, logo depois de ler apenas o título ou o endereço de origem.
- O usuário abre a mensagem e fica algum tempo (provavelmente) lendo-a.
- O usuário abre, lê e armazena (ou apaga) uma mensagem.
- O usuário abre, lê e responde (ou encaminha) uma mensagem para um ou mais usuários.

Pode ser observado, nestes cenários simplificados (mas representativos do mundo real), que qualquer tarefa automatizada (a ser feita em benefício do usuário) tem que lidar com incerteza. Por exemplo, tarefas como sugerir novos recursos ao usuário (por exemplo, páginas *Web*, mensagens de correio eletrônico ou grupos de notícias), ou tarefas de organização do ambiente da aplicação do usuário (por exemplo, organização das caixas de correio eletrônico). A incerteza é inerente ao processo de aprendizagem dos padrões de comportamento do usuário e também ao processo de inferência das preferências ou intenções dos usuários. Por esta razão, o problema de como modelar e trabalhar com conhecimento incerto é focalizado neste trabalho.

1.2 Uma Breve Revisão de Algumas Abordagens Anteriores

O problema da representação de conhecimento para raciocinar mediante incerteza tem sido abordado desde o começo da teoria da probabilidade, séculos atrás. Diferentes técnicas e formalismos foram desenvolvidas desde então para lidar com informação incompleta e/ou incerta, sendo que [HUN 98] classificou-as como numéricas e simbólicas, sendo aqui descritas de forma resumida as características mais interessantes no contexto deste trabalho.

1.2.1 Abordagens Numéricas

As principais teorias formais que foram desenvolvidas para tratar incerteza são a Teoria da Probabilidade (cujos fundamentos básicos são apresentados no próximo capítulo), Teoria da Evidência [Dempster, 1968; Shafer, 1976 Apud FUN 86 and RIC 94], Teoria da Possibilidade [ZAD 78] e Fatores de Certeza [DAV 77]. Segundo [HUN 98], estas teorias são similares, diferindo apenas em detalhes de significados ou de aplicação. A crença de cada teoria pode ser associada com base na informação estatística, possibilidade física, ou uma avaliação subjetiva de um especialista. Dado o resultado de uma distribuição de crença, o interesse está em como as crenças associadas podem ser manipuladas. As Teoria da Probabilidade e a Teoria da Evidência limitam o total de crença que pode ser determinado por uma função de distribuição através da restrição de que a soma de todas as crenças deve ser 1.

A Teoria da Evidência foi desenvolvida por [Dempster, 1968; Shafer, 1976 Apud FUN 86 and RIC 94] para contornar o que eles consideravam como limitações da Teoria da Probabilidade. A Teoria da Possibilidade é uma teoria formal de possibilidades que foi introduzida por [ZAD 78], sendo baseada na noção de conjunto *fuzzy*. Existe uma conexão heurística entre Possibilidade e Probabilidade, uma vez que algo impossível tende também a ser algo improvável, mas um alto grau de Possibilidade não implica num alto grau de Probabilidade, tampouco um baixo grau de Probabilidade implica em um baixo grau de Possibilidade.

Fatores de Certeza [DAV 77] foram amplamente usados para lidar com incerteza por serem simples e intuitivos. A abordagem de Fator de Certeza designa um peso numérico (o Fator de Certeza) para as regras do sistema. Todavia, [HEC 86] demonstrou que a definição original do modelo de Fatores de Certeza falha, uma vez que a crença em uma hipótese (dado duas evidências) vai depender da ordem na qual o efeito destas evidências é computado.

1.2.2 Abordagens Simbólicas

As lógicas não-monotônicas foram introduzidas para permitir programas a lidar com conhecimento incompleto através do uso de raciocínio de “senso comum”, evitando desta forma a necessidade de declarar cada possível exceção para uma regra geral. Lógica *Default* [REI 80], Argumentação (*Argumentation*) [94; Apud PAR 98] e Sistemas Mantenedores de Verdade (*Truth Maintenance Systems*) [DOY 79, KLE 86] estão entre as principais abordagens simbólicas. A motivação comum a todos estes sistemas não-monotônicos é superar as limitações da lógica de primeira ordem, sendo basicamente dois problemas principais: i) Qualificação: não existe um quantificador do tipo “geralmente”; e ii) Monotonicidade: não existe mecanismo para desfazer inferências uma vez que elas são realizadas, ou remover fatos uma vez que eles foram acrescentados à base de conhecimento.

A Lógica *Default* [REI 80] modela raciocínios prototipados através do uso de regras de inferência especiais (conhecidas como regras *default*) acrescentadas na lógica de primeira ordem padrão.

A Argumentação [94; Apud PAR 98] é um processo que constrói e compara argumentos. Um argumento pode ser estruturado a partir de fatos de tal forma que uma afirmação (uma conclusão) pode ser argumentada (inferida) se e somente se: i) existe alguma garantia (algumas suposições adicionais) que podem ser usadas com os fatos para derivar logicamente a afirmação; e ii) não existe nenhum outro argumento que possa refutar a afirmação.

Um Sistema Mantenedor de Verdade grava informação sobre cada inferência que é gerada a partir de um conjunto de suposições. Os dois principais tipos deste sistema são o Sistema Mantenedor de Verdade baseado em Justificativa [DOY 79] e o Sistema Mantenedor de Verdade baseado em Suposição [KLE 86]. O primeiro grava um único conjunto de fatos consistentes e todas as inferências que podem ser provadas a partir deles. Um sistema externo (que pode ser o usuário) é chamado para resolver toda inconsistência que é detectada e então o Sistema Mantenedor desfaz as necessárias inferências. Já o segundo tipo deste sistema mantenedor mantém todos os subconjuntos consistentes de um conjunto de fatos conhecidos e mantém também todas as inferências que podem ser construídas a partir de cada fato. Neste sistema a inconsistência é tratada através da criação de novos subconjuntos consistentes e da identificação de quais inferências podem ser construídas a partir deles.

Em termos de lógica *default*, um Sistema Mantenedor de Verdade baseado em Justificativa pode ser considerado como um meio de estabelecer uma extensão única e um Sistema Mantenedor de Verdade baseado em Suposição como um meio de estabelecer todas as possíveis extensões.

1.2.3 Observações sobre as Abordagens

Algumas classes de problemas requerem a combinação de diferentes formalismos (por exemplo, combinando técnicas simbólicas com técnicas numéricas) para modelar diferentes aspectos de incerteza presentes num certo domínio de aplicação.

Acerca do uso das técnicas numéricas, existe a dificuldade de obter os números, visto que nem todos os domínios provêm os dados estatísticos necessários para dar suporte ao enfoque frequencalista da teoria da probabilidade. Outra dificuldade está no uso dos números, ou seja, na interpretação dos resultados. Para uma interpretação mais precisa, torna-se necessário associar semântica aos dados, identificando o tipo que crença que eles representam numericamente.

Apesar destes problemas de obtenção dos números e de interpretação de resultados (além da necessidade de escolher um formalismo ou combinar diversos), o interesse de pesquisadores e da indústria de software têm crescido na aplicação de representações estatísticas baseadas nas relações de dependência condicional e também no uso de técnicas de inferência estatística baseadas na Teoria da Probabilidade, conforme pode ser visto em algumas das principais conferências da área de inteligência artificial (por exemplo, www.auai.org).

Redes Bayesianas (ou redes de crença) [PEA 88] são usadas para modelar situações nas quais causalidade, ou influência predominam, mas são situações onde se tem apenas um entendimento parcial, portanto existe a necessidade de modelar probabilisticamente. As redes Bayesianas representam e raciocinam mais eficientemente com informação probabilística através do uso de probabilidades condicionais acrescidas de informação estrutural extra. Mais especificamente, as relações de independência condicional são usadas para diminuir a dimensionalidade e o número de declarações de probabilidade condicional, simplificando assim os cálculos.

Considerando a motivação deste trabalho (seção 1.1), além do fato da aprendizagem de padrões de comportamento e da inferência das intenções e preferências dos usuários serem processos onde incerteza é intrínseca [HOR 97], desta forma no capítulo 2 serão revistos os principais conceitos que fundamentam o desenvolvimento desta proposta de tese.

1.2.4 As Principais Limitações das Abordagens Anteriores

Dentre as principais limitações das abordagens anteriores, pode ser antecipado neste capítulo inicial que as raízes das abordagens anteriores pertencem a um paradigma no qual existe uma lacuna entre o processo de aprendizagem e o processo de decisão/execução. Esta lacuna decorria da dificuldade de organizar, armazenar e atualizar grandes volumes de dados de uma forma eficiente, contínua e incremental. O problema da lacuna geralmente inclui: preparação dos dados, descoberta de padrões, avaliação do conhecimento, refinamento, etc. No caso específico de redes Bayesianas, o processo de aprendizagem é uma instância de um problema NP-HARD [CHI 94]. Embora tenham surgido técnicas mais aperfeiçoadas (por exemplo, [CAS 97, JEN 2001]), elas ainda sofrem do problema da lacuna – que causa a perda de uma grande quantidade de relacionamentos semânticos (dos contextos de origem dos dados) enquanto os dados estão sendo massivamente armazenados, antes da execução do processo de aprendizagem. Esta perda de relacionamentos semânticos cria um *trade-off* entre coleta de dados e rastreamento, ou seja, *uma* dificuldade de conciliar as necessidades de: i) Coletar dados suficientes (para alcançar importância estatística); versus ii) Rastrear (ou acompanhar) comportamentos não-estacionários (por exemplo, usando técnicas de esmaecimento (*fading*)).

Além disso, a redução da lacuna requer um sistema que aprenda durante a execução – o qual precisa do suporte de uma eficiente infraestrutura que tenha uma forte organização contextual. Esta organização deve permitir o controle da semântica e das relações de causa e efeito entre as percepções dos objetos do ambiente, percepções das ações (realizadas sobre, ou relacionadas com os objetos do ambiente) e percepções da realimentação (*feedback*) dos resultados/efeitos das ações. Adicionalmente, se for considerado um ambiente dinâmico, complexo e incerto, existe a necessidade de integrar o processo de percepção do ambiente, tomada de decisão e aprendizagem, porque somente um sistema completamente integrado pode obter a sinergia necessária para lidar com as dificuldades relacionadas.

1.3 O Conteúdo desta Proposta de Tese

Alguns fundamentos básicos de probabilidades e redes probabilísticas são revisados no capítulo 2. Em seguida, este trabalho apresenta no capítulo 3 os Serviços para Auxiliar Decisão mediante Incerteza (SADI) contendo um modelo de representação de conhecimento que é eficiente por ser escalável (*scaleable*) e construído de forma contínua e incremental a partir das relações de causa e efeito entre percepções e ações. Assume-se que é possível perceber o ambiente, é necessário tomar decisões mediante incerteza, e é possível perceber a realimentação (*feedback*) referente ao sucesso ou fracasso das ações escolhidas, e também assume-se que é possível aprender com a experiência.

Uma vez que a representação de conhecimento do SADI é organizada em vários contextos, quando um padrão percebido ocorre em mais de um contexto, ele é eficientemente replicado e distribuído (através de operações incrementais que permitem crescimento contínuo). Um padrão percebido que é reconhecido num contexto é definido aqui como um símbolo. Cada contexto, sempre que ativado, realiza sua própria contagem lógica de tempo – que é usado para associar probabilidades às percepções. As percepções entradas no sistema em um dado instante tem uso duplo: elas alimentam adiante (*feedforward*) a ativação de contextos e elas também realimentam (*feedback*) os possíveis efeitos/resultados das ações prévias. Como cada contexto possui sua própria

contagem lógica de tempo, sua própria semântica e suas próprias replicas de símbolos, então a relação de causa e efeito é relativa a cada contexto ativado.

O ciclo de operação do *Framework SADI* pode ser brevemente descrito assim: 1.o) Percepções são entradas; 2.o) Um ou mais Contextos são ativados; 3.o) Percepções são reconhecidas; 4.o) O *feedback* é processado, dado as percepções reconhecidas nos contextos ativados; 5.o) Ações são decididas e realizadas, dado os contextos ativados. A operação do SADI é cíclica porque as percepções induzem ações e as ações produzem novas percepções. Apesar da descrição seqüencial do passo 1 até o passo 5, o processamento destes passos pode ser implementado como um *pipeline*, e os passos 4 e 5 podem ser executados em paralelo se eles ocorrem em contextos diferentes. Este processamento paralelo pode ser executado mesmo se os diferentes contextos possuem símbolos em comum, porque cada contexto possui suas próprias réplicas de símbolos, sua própria semântica, e suas própria contagem lógica do tempo.

Com relação ao serviço de detecção de contexto do SADI, ele provê dois serviços básicos: i) O casamento sintático e semântico dos símbolos percebidos; e ii) a eficiente organização de símbolos compostos mais complexos semanticamente. Mesmo com o uso dos símbolos compostos, o *Framework SADI* ainda preserva as propriedades das operações incrementais que permitem crescimento contínuo.

Além de reduzir a lacuna entre o processo de aprendizagem e o processo de decisão/execução, o SADI também minimiza a perda semântica do *trade-off* (entre coletar dados suficientes *versus* rastrear comportamentos não-estacionários), porque as relações de causa e efeito estão organizadas em contextos.

No capítulo 4, alguns dos principais conceitos definidos no *Framework SADI* são ilustrados através da apresentação de um estudo de caso sobre um agente de software que monitora o comportamento do usuário e constrói um modelo sobre seu comportamento e preferências, visando auxiliá-lo no consumo de informação via Canais de Informação na Internet. Finalmente, o capítulo 5 apresenta as principais conclusões e contribuições deste trabalho.

2 CONCEITOS BÁSICOS E ESTADO DA ARTE

Um dos modelos mais usados para modelar e inferir o comportamento humano é o de Redes Probabilísticas (ou Bayesianas), o qual é baseado na teoria da probabilidade. Sendo assim, ao longo deste capítulo vamos expor os fundamentos básicos da teoria da probabilidade e das Redes Probabilísticas, fazendo depois uma análise das suas características e também a evolução do estado da arte, observando as deficiências e limitações que precisam ser resolvidas.

2.1 Introdução à Probabilidade

A origem da teoria da probabilidade é atribuída por [DEL 74] aos estudos de jogos de azar realizados por Pascal e Fermat (1654), Huygens (1656) e Jacques Bernoulli (o qual menciona a lei dos grandes números em 1713). Ainda segundo [DEL 74], em 1662 Graunt fez estudos probabilísticos sobre a mortalidade humana, e em 1693 Halley desenvolveu aplicações para seguros de vida. A teoria da probabilidade continuou se desenvolvendo no século XVIII com os trabalhos de Daniel Bernoulli sobre as probabilidades contínuas. Contribuíram depois Lagrange, Laplace e Gauss, com uma teoria dos erros que resultou o método dos mínimos quadrados. A partir de então, inúmeras contribuições se sucederam, dando suporte para os mais recentes modelos e técnicas usados pela área de Inteligência Artificial (I.A.) para representar conhecimento incerto e fazer tomada de decisão mediante incerteza – que constitui o tema principal deste trabalho de pesquisa.

Os estudos clássicos da teoria de probabilidade lidam com experimentos aleatórios ou não-determinísticos onde são definidos modelos matemáticos para associar probabilidades aos eventos relacionados com os experimentos. Em [LIP 72,p.53] se encontram algumas definições básicas: **Espaço Amostral** é “o conjunto S de todos os resultados possíveis de um experimento”. **Ponto Amostral** é “um resultado particular, isto é, um elemento de S . Um **Evento** A é “um conjunto de resultados ou, em outras palavras, um subconjunto do espaço amostral S .” Ainda em [LIP 72,p.53] estão definidos que o conjunto vazio \emptyset e o espaço amostral S são eventos, onde \emptyset é chamado de evento impossível e S de evento certo. Os eventos ainda podem ser combinados através de operações com conjuntos:

“(i) o evento $A \cup B$ ocorre se e somente se ocorre A ou ocorre B (ou ambos);

(ii) o evento $A \cap B$ ocorre se e somente se ocorrem A e B ;

(iii) o evento A^C , complemento de A , ocorre se e somente se não ocorre A .

Dois eventos A e B são ditos mutuamente exclusivos se são disjuntos, isto é, se $A \cap B = \emptyset$ não ocorrem simultaneamente.”

A teoria da probabilidade possui duas principais interpretações acerca do conceito de probabilidade. A primeira, também chamada de enfoque frequencista, considera que o valor numérico da probabilidade deve ser encontrado com base na experimentação, sendo este enfoque baseado na lei dos grandes números mencionada por Bernoulli em 1713: “É muito pouco provável que, se efetuarmos um número suficientemente grande de experimentos, a frequência relativa de um acontecimento se afaste muito da sua probabilidade” [LOP 99,p.67]. Quanto à segunda interpretação do conceito de probabilidade, existe o enfoque subjetivista ou Bayesiano, muito utilizado na área de I.A. depois dos trabalhos de [PEA 88] sobre raciocínio probabilístico. A explicação da interpretação Bayesiana também é apresentada em [PEA 2000,p.2] como sendo uma interpretação de probabilidade na qual os valores numéricos codificam o grau de crença sobre eventos do mundo, e os dados (coletados do domínio) são usados para reforçar, atualizar ou enfraquecer estes graus de crença. No formalismo em questão, os graus de crença são associados a proposições em alguma linguagem e depois são combinados e manipulados de acordo com as regras do cálculo de probabilidades.

Independente da interpretação ou enfoque, [LOP 99,p.67] declara que a teoria da probabilidade se baseia nos axiomas estabelecidos pelo matemático russo Kolmogorov. “Em um experimento aleatório com um espaço amostral associado, uma função que associa a cada resultado um número real, representado por $P(A)$, é uma função de probabilidade se satisfaz às seguintes propriedades:

- 1) Para qualquer evento A , $0 \leq P(A) \leq 1$;
- 2) A probabilidade do evento certo é 1, $P(\text{Espaço Amostral})=1$;
- 3) Se os eventos A e B são incompatíveis (mutuamente excludentes), a probabilidade de ocorrência do evento A ou do evento B é a soma da probabilidade de ocorrência de A com a probabilidade de ocorrência de B .”

Alguns teoremas decorrem diretamente destes axiomas, cujas provas são extraídas de [LIP 72,p.55]:

- “Se \bar{A} é o conjunto vazio, então $P(\bar{A})=0$. Prova: Seja A um conjunto qualquer; então A e \bar{A} são disjuntos e $A \cup \bar{A} = A$.” Pelo 3.o axioma:

$$P(A) = P(A \cup \bar{A}) = P(A) + P(\bar{A})$$
 Subtraindo $P(A)$ de ambos os lados, temos nosso resultado.”
- “Se A^C é o complemento de um evento A , então $P(A^C)=1-P(A)$. Prova: O espaço amostral S pode ser decomposto nos eventos mutuamente exclusivos A e A^C ; ou seja, $S = A \cup A^C$.” Pelos axiomas 2 e 3, se obtem

$$1 = P(S) = P(A \cup A^C) = P(A) + P(A^C),$$
 de onde vem o nosso resultado”.
- “Se $A \hat{I} B$, então $P(A) \leq P(B)$. Prova: Se $A \hat{I} B$, então B pode ser decomposto nos eventos mutuamente exclusivos A e $B \setminus A$ ” (vide Figura 2.1-i). “Assim,

$$P(B) = P(A) + P(B \setminus A)$$
 O resultado segue então, do fato de $P(B \setminus A) \geq 0$.”

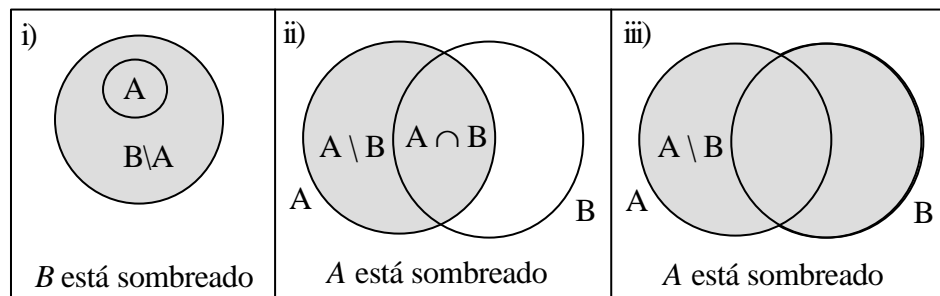


Figura 2.1: Relações de Conjuntos
Adaptação de [LIP 72,p.55]

- “Se A e B são dois eventos quaisquer, então $P(A \setminus B) = P(A) - P(A \cap B)$. Prova: Ora, A pode ser decomposto nos eventos mutuamente exclusivos $A \setminus B$ e $A \cap B$; ou seja,

$$A = (A \setminus B) \dot{\cup} (A \cap B) \quad (\text{vide Figura 2.1-ii})$$

Portanto, pelo axioma 3,

$$P(A) = P(A \setminus B) + P(A \cap B),$$

de onde vem o nosso resultado.”

- “Se A e B são dois eventos quaisquer, então

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Prova: Note que $A \cup B$ pode ser decomposto nos eventos mutuamente exclusivos $A \setminus B$ e B ; ou seja, $A \cup B = (A \setminus B) \cup B$.” Desta forma, pelo axioma 3 e pela prova anterior $P(A \setminus B) = P(A) - P(A \cap B)$:

$$P(A \cup B) = P(A \setminus B) + P(B)$$

$$= P(A) - P(A \cap B) + P(B)$$

$$= P(A) + P(B) - P(A \cap B) \quad \text{que é o resultado desejado.}”$$

2.2 Variável Aleatória e Distribuição de Probabilidades

Dado um experimento, deseja-se saber a probabilidade de ocorrência de determinadas características dos resultados possíveis, sendo que estas características podem ter valores numéricos para representá-las. Seja X a função que associa um resultado de um experimento a um número, esta função é chamada de **Variável Aleatória**, e [LOP 99,p.73] a apresenta como sendo uma “função real definida em um espaço amostral cujo domínio é o conjunto de resultados e cuja imagem é o conjunto de números reais”. Um comentário interessante é apresentado por [LOP 99,p.73] sobre o nome ‘Variável Aleatória’ (o qual é universalmente aceito), todavia X não é uma variável mas sim uma função cujo resultado é aleatório, mas não os valores da função.

Com relação à distribuição de probabilidades, [LOP 99,p.73] compara a construção de tabelas de frequências sem perda de informação (na estatística descritiva) com a construção de tabelas que associam valores com suas probabilidades de ocorrência (no cálculo de probabilidades). Mais especificamente, **Distribuição de Probabilidade** é “o conjunto de todas as possibilidades de um experimento aleatório e suas probabilidades de ocorrência”. A construção de uma tabela de Distribuição de Probabilidades é ilustrada neste exemplo extraído de [LOP 99,p.73]: Um certo equipamento possui uma taxa de 80% de reprovação, e em cada experimento três equipamentos são testados (sendo que o teste de cada equipamento não depende do

resultado do teste de outro equipamento). Desta forma, deseja-se construir a distribuição de probabilidade do número X de equipamentos que são reprovados. A Tabela 2.1 associa cada resultado a um número (onde R significa equipamento reprovado e A , aprovado).

Tabela 2.1: Resultados dos testes com 3 equipamentos

Resultado elementar	X
RRR	3
RRA	2
RAR	2
ARR	2
RAA	1
ARA	1
AAR	1
AAA	0

Fonte: [LOP 99,p.73]

Em cada um dos resultados elementares calcula-se a probabilidade de ocorrência considerando a taxa de 80% de reprovação de cada teste. A Tabela 2.2 exibe os cálculos.

Tabela 2.2: Probabilidades dos Resultados dos Testes

Resultado elementar	X	Probabilidade
RRR	3	$0,8 \times 0,8 \times 0,8 = 0,512 = 51,2\%$
RRA	2	$0,8 \times 0,8 \times 0,2 = 0,128 = 12,8\%$
RAR	2	$0,8 \times 0,2 \times 0,8 = 0,128 = 12,8\%$
ARR	2	$0,2 \times 0,8 \times 0,8 = 0,128 = 12,8\%$
RAA	1	$0,8 \times 0,2 \times 0,2 = 0,032 = 3,2\%$
ARA	1	$0,2 \times 0,8 \times 0,2 = 0,032 = 3,2\%$
AAR	1	$0,2 \times 0,2 \times 0,8 = 0,032 = 3,2\%$
AAA	0	$0,2 \times 0,2 \times 0,2 = 0,008 = 0,8\%$

Fonte: [LOP 99,p.73]

A Tabela 2.3 mostra os valores da distribuição de probabilidades da variável aleatória X , resultantes dos valores da Tabela 2.2.

Tabela 2.3: Distribuição de probabilidades da variável aleatória X

X	Probabilidade
0	$0,8\% = 0,008$
1	$9,6\% = 0,096 = 0,032 + 0,032 + 0,032$
2	$38,4\% = 0,384 = 0,128 + 0,128 + 0,128$
3	$51,2\% = 0,512$
Total	$100\% = 1$

Fonte: [LOP 99,p.73]

Considerando ainda [LOP 99,p.73], onde as letras minúsculas x_1, x_2, \dots, x_n são usadas como notação dos resultados de uma variável aleatória X , uma função de probabilidade $P(x_i)$ tem as seguintes propriedades:

- a. $P(x_i) \geq 0$ para todo i , onde $P(x_i) = P(X = x_i)$, $i = 1, 2, \dots$
- b. $\sum_{\text{para todo } i} P(x_i) = 1$.

Se a distribuição de probabilidades de uma variável aleatória é explicitamente conhecida, então todo o resumo estatístico (por exemplo, média e desvio padrão) também será conhecido. Na Estatística Descritiva, também se construiu uma tabela de freqüências com perda de informação, na qual uma freqüência absoluta (e também uma freqüência relativa) é associada a cada classe. Ao se fazer o mesmo com relação a um intervalo de valores e suas probabilidades, origina-se uma tabela que associa a cada intervalo a sua probabilidade de ocorrência.” [LOP 99,p74-75].

A Distribuição de Probabilidades é definida em [CAS 97,p.72-74] também considerando as notações $\{X_1, \dots, X_n\}$ como um conjunto de variáveis aleatórias e $\{x_1, \dots, x_n\}$ como um conjunto de suas possíveis instâncias (por exemplo, se X_i é uma variável binária, x_i pode ser 1 ou 0). Sendo assim, $P(x_1, \dots, x_n)$ representa a **Distribuição de Probabilidade Conjunta (DPC)** das variáveis em X , isto é

$$P(x_1, \dots, x_n) = P(X_1 = x_1, \dots, X_n = x_n)$$

Por exemplo, a Tabela 2.4 apresenta uma Distribuição de Probabilidade Conjunta (DPC) para três variáveis binárias A, B e C .

Tabela 2.4: Exemplo de Tabela de Distribuição de Probabilidade Conjunta (DPC)

A	B	C	DPC
0	0	0	0,225
0	0	1	0,045
0	1	0	0,315
0	1	1	0,205
1	0	0	0,125
1	0	1	0,045
1	1	0	0,035
1	1	1	0,005

Ainda em [CAS 97,p.72-74], a **Distribuição de Probabilidade Marginal (DPM)** de uma variável x_i é obtida pelo seguinte cálculo:

$$P(x_i) = P(X_i = x_i) = \sum_{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n} P(x_1, \dots, x_n)$$

Para ilustrar o cálculo de uma Distribuição de Probabilidade Marginal, são usados agora os valores da Tabela 2.4, para poder saber $P(B=1)$. Portanto, deve-se somar os valores da coluna DPC (que representa a Distribuição de Probabilidade Conjunta) onde $B=1$, isto é, o valor da variável B é fixado em 1 e o somatório é feito variando todos os estados das variáveis A e C . A Tabela 2.5 ilustra este processo do cálculo de $P(B=1)$.

Tabela 2.5: Exemplo de Cálculo de Probabilidade Marginal

A	B	C	DPC
0	0	0	0,225
0	0	1	0,045
0	1	0	0,315
0	1	1	0,205
1	0	0	0,125
1	0	1	0,045
1	1	0	0,035
1	1	1	0,005

$$P(B=1) = 0,315 + 0,205 + 0,035 + 0,005$$

$$= \mathbf{0,56}$$

Com relação ao cálculo da Distribuição de Probabilidade Marginal, o uso do somatório se aplica ao caso das variáveis discretas, mas [CAS 97,p.72-74.] expõe que no caso de variáveis contínuas, o somatório é substituído por uma integral.

De qualquer forma, seja uma Variável Aleatória X discreta ou contínua, existe um conceito similar ao da média ponderada que é o Valor Esperado. Conforme descrito em [LOP 99,p.79], “uma distribuição de probabilidades é semelhante a uma distribuição de frequências sem perda de informação”. Portanto, o cálculo do **Valor Esperado** de uma Variável Aleatória X - denotado por $E(X)$ ou μ_X , é similar ao cálculo da média ponderada. Isto é exposto de uma forma simples em [LIP 72,p.110], onde $E(X)$ é a média ponderada dos possíveis valores de X (x_1, x_2, \dots, x_n), sendo cada um ponderado pela sua função de probabilidade $f(x_i)$:

$$E(X) = x_1 \cdot f(x_1) + x_2 \cdot f(x_2) + \dots + x_n \cdot f(x_n) \quad \text{ou seja, } E(X) = \sum_i x_i \cdot f(x_i)$$

Segundo [LOP 99,p.79], esta função de probabilidade $f(x_i)$ é a distribuição de probabilidade $P(x_i)$ se X é discreta. Quando X é contínua, o cálculo do Valor Esperado é obtido por

$$E(X) = \int_{-\infty}^{\infty} x \cdot f(x) dx$$

onde $f(x)$ é uma função de densidade de probabilidade.

Quando existe uma tabela de Distribuição de Probabilidade Conjunta para uma coleção de Variáveis Aleatórias, [NIL 98] afirma que é possível calcular todas as (Distribuições de) Probabilidade Marginal e todas as (Distribuições de) Probabilidade Conjunta de ordem mais baixa. Todavia, [NIL 98] aponta um problema: quando o domínio da aplicação envolve um grande conjunto de Variáveis Aleatórias, a construção desta tabela torna-se praticamente intratável.

2.3 Probabilidade Condicional e Regra de Bayes

O conceito de probabilidade condicional é ilustrado através de um exemplo apresentado em [RUS 95,p.425]. Considere a Distribuição de Probabilidade Conjunta (DPC) sobre cárie e dor de dente na Tabela 2.6. A probabilidade de ter dor de dente $P(\text{Dor_Dente} = v)$ é obtida pelo cálculo da probabilidade marginal:

$$\begin{aligned} P(\text{Dor_Dente} = v) &= \sum_{\text{Cárie}} P(\text{Dor_Dente}=v, \text{Cárie}) \\ &= P(\text{Dor_Dente}=v, \text{Cárie}=F) + P(\text{Dor_Dente}=v, \text{Cárie}=T) \\ &= 0,01 + 0,04 = 0,05 \end{aligned}$$

Tabela 2.6: Exemplo de DPC sobre Cárie e Dor de Dente

Dor_Dente	Cárie	DPC
F	F	0,89
F	V	0,06
V	F	0,01
V	V	0,04

Adaptação de [RUS 95]

A probabilidade de ter dor de dente e cárie ao mesmo tempo já está representada diretamente na tabela de DPC: $P(\text{Dor_Dente}=T, \text{Cárie}=T) = 0,04$.

A partir de uma evidência (p.ex., Dor_Dente), é possível fazer uma inferência probabilística a respeito da existência de outra Variável Aleatória (p.ex., Cárie). Isto é feito através do cálculo da **Probabilidade Condicional**:

$$\begin{aligned} P(\text{Dor_Dente} =_{=T} | \text{Cárie} =_{=T}) &= \frac{P(\text{Dor_Dente} =_{=T}, \text{Cárie} =_{=T})}{P(\text{Dor_Dente} =_{=T})} \\ &= 0,04 / 0,05 = 0,80. \end{aligned}$$

Em outras palavras, a probabilidade condicional de ter dor de dente (D) dada a evidência de cárie (C), isto é $P(D/C)$, é a probabilidade de $P(D,C)$ dividido por $P(C)$. Ou seja, a probabilidade de ter dor de dente $P(\text{Dor_Dente} = T)$ aumentou de 0,05 para 0,80 depois da evidência de cárie $P(\text{Dor_Dente}=T | \text{Cárie}=T)$.

Ao contrário da definição tradicional de Probabilidade Condicional baseada em termos de Distribuição de Probabilidade Conjunta, ou eventos conjuntos (*joint events*) $P(A/B) = P(A,B)/P(B)$, [PEA 2000,p.3-4] declara que os filósofos Bayesianos consideram o relacionamento condicional como sendo mais básico (do que aqueles de eventos conjuntos) e mais compatível com a organização do conhecimento humano. Nesta visão Bayesianana, B serve como um ponteiro para um contexto ou quadro (*frame*) de conhecimento, e A/B significa o evento A no contexto especificado por B (por exemplo, o sintoma A no contexto de uma doença B). Portanto, o conhecimento empírico poderá ser codificado em declarações de probabilidade condicional $P(A/B)$, assim como crenças em eventos conjuntos $P(A, B)$ poderão ser obtidos destas declarações através da **Regra do Produto** $P(A, B) = P(A/B).P(B)$.

Conforme [PEA 2000,p.3-4], as expressões básicas do formalismo Bayesiano são declarações sobre probabilidades condicionais (o que envolve declarações de dependência e independência condicional). A definição de **Independência Condicional** a ser apresentada agora é baseada em [CAS 97,p.72-74].

Sejam X e Y dois subconjuntos disjuntos, oriundos do conjunto de Variáveis Aleatórias $\{X_1, X_2, \dots, X_n\}$. Então X é dito ser independente de Y se e somente se

$$P(x/y) = P(x)$$

para todos os valores possíveis de x e y das variáveis X e Y ; caso contrário, X é dito ser dependente de Y . Isto significa que se X é independente de Y , então o conhecimento sobre Y não afeta o conhecimento sobre X . Se X é independente de Y $P(x/y) = p(x)$, então a relação de Dependência Condicional $P(x/y) = P(x,y)/P(y)$ se reduz a

$$P(x,y) = P(x). P(y)$$

Uma outra situação de independência condicional é apresentada por [PEA 2000,p.3-4]: se $P(x/y,z) = P(x/z)$ então X e Y são condicionalmente independentes, porque uma vez que se conhece z , o conhecimento de y não altera a crença em x . Para concluir, [CAS 97,p.72-74] afirma que a relação de independência condicional é simétrica, isto é, se X é independente de Y , então Y é também independente de X .

Com relação à **Regra de Bayes**, em [RUS 95,p.425] pode ser visto que ela decorre das relações de probabilidade condicional. Por um lado,

$$P(A, B) = P(A / B). P(B)$$

mas também

$$P(A, B) = P(B / A). P(A)$$

Ao se igualar os dois lados à direita de $P(A, B)$, e depois dividindo por $P(A)$ se obtém a conhecida Regra de Bayes:

$$P(B | A) = \frac{P(A | B).P(B)}{P(A)}$$

Um exemplo do uso da Regra de Bayes, oriundo de [RUS 95,p.425], mostra a importância prática desta regra numa situação de diagnóstico médico, onde existem probabilidades condicionais sobre relacionamentos causais e se deseja derivar um diagnóstico. Um médico sabe que meningite (M) causa enrijecimento de pescoço (EP) em 50% dos pacientes, isto é, $P(EP/M) = 0,5$. Esta probabilidade, $P(EP/M)$, também é denominada *likelihood* (possibilidade, ou verosimilhança [LAD 2000]). Este médico também sabe que a probabilidade de contrair meningite $P(M)$, à *priori*, é de 1/50000. Além disso, ele sabe que a probabilidade de ter enrijecimento de pescoço $P(EP)$, à *priori*, é de 1/20. Portanto, quando um novo paciente apresenta o sintoma de enrijecimento de pescoço, este médico calcula a probabilidade do paciente ter meningite da seguinte forma:

$$P(M | EP) = \frac{P(EP | M) \times P(M)}{P(EP)} = \frac{0,5 \times 1 / 50000}{1 / 20} = 1 / 5000$$

A interpretação deste resultado é que apenas 1 (em cada 5000 pacientes com pescoço rígido) está com meningite. Ainda [RUS 95,p.425] argumenta que esta abordagem é mais flexível do que manter de forma direta a probabilidade condicional $P(M|EP) = P(M, EP) / P(EP)$. Esta flexibilidade decorre do fato da probabilidade à

priori (também denominada como probabilidade incondicional) de meningite $P(M)$ poder ser atualizada caso ocorra um surto de meningite, e que então $P(M/EP)$ crescerá na proporção de $P(M)$ usando a Regra de Bayes. Caso $P(M/EP)$ fosse calculado pela outra forma (a direta), esta atualização não seria possível.

2.4 Regra da Cadeia, Diagrama de Árvore e Teorema de Bayes

A Regra do Produto $P(A_1, A_2) = P(A_1/A_2).P(A_2)$ serve de base para o que [PEA 88,p.31-32] considerou uma generalização útil, muito conhecida como **Regra da Cadeia** (também chamada de Teorema da Multiplicação para probabilidade condicional [LIP 72,p.78-82]). O processo de generalização ocorre por indução, substituindo o conjunto unitário (A_i) pelo conjunto (A_i, A_{i+1}):

$$P(A_1, A_2) = P(A_1/A_2).P(A_2) \rightarrow P(A_1, A_2, A_3) = P(A_1/A_2, A_3).P(A_2, A_3)$$

Aplica-se a Regra do Produto no último termo $P(A_i, A_{i+1})$ e assim se obtém:

$$P(A_1, A_2, A_3) = P(A_1/A_2, A_3).P(A_2, A_3) \rightarrow P(A_1, A_2, A_3) = P(A_1/A_2, A_3).P(A_2/A_3).P(A_3)$$

Portanto, de uma forma genérica, a Regra da Cadeia pode ser expressa da seguinte forma:

$$P(A_1, A_2, \dots, A_n) = P(A_1/A_2, \dots, A_n) \dots P(A_{n-1}/A_n).P(A_n)$$

Para exemplificar o conceito do teorema da (Regra da Cadeia), [LIP 72,p.78-82] apresenta este exemplo: “Num lote de 12 peças, 4 são defeituosas. Três peças são retiradas aleatoriamente, uma após a outra. Encontre a probabilidade P de todas essas três peças serem não defeituosas.

A probabilidade da primeira peça ser não defeituosa é $8/12$, já que 8 das 12 peças são não defeituosas. Se a primeira peça é não defeituosa, então a probabilidade da próxima ser não defeituosa é $7/11$, pois somente 7 das 11 peças restantes são não defeituosas. Se as duas primeiras são não defeituosas, então a probabilidade da última ser não defeituosa é $6/10$, já que somente 6 das 10 restantes são não defeituosas. Então, pelo Teorema da Multiplicação:”

$$P = \frac{8}{12} \times \frac{7}{11} \times \frac{6}{10} = \frac{14}{55}$$

Ou seja, se $P(A, B, C)$ é a probabilidade das peças A, B e C serem não defeituosas, Então, na expressão $P(A, B, C) = P(A/B, C).P(B/C).P(C)$, $P(C) = 8/12$, $P(B/C) = 7/11$ e $P(A/B, C) = 6/10$.

Ainda em [LIP 72,p.78-82], o conceito de **Processo Estocástico** (finito) é apresentado como sendo “uma seqüência (finita) de experimentos, na qual cada experimento tem um número finito de resultados, com uma dada atribuição de probabilidade”. O **Diagrama de Árvore** é usado para descrever um Processo Estocástico e calcular a probabilidade de eventos. O seguinte exemplo ilustra estes conceitos, onde “são dadas três caixas, como segue:

A caixa I tem 10 lâmpadas, das quais 4 são defeituosas.

A caixa II tem 6 lâmpadas, das quais 1 é defeituosa.

A caixa III tem 8 lâmpadas, das quais 3 são defeituosas.”

O problema consiste em saber a probabilidade P de uma lâmpada ser defeituosa, ao se selecionar uma caixa aleatoriamente e depois selecionar uma lâmpada também aleatoriamente. Assim, uma seqüência de dois experimentos foi realizada: 1.o) Uma das três caixas é selecionada; e 2.o) É selecionada uma lâmpada defeituosa (D) ou não (N). A Figura 2.2 mostra o Diagrama de Árvore que descreve este processo.

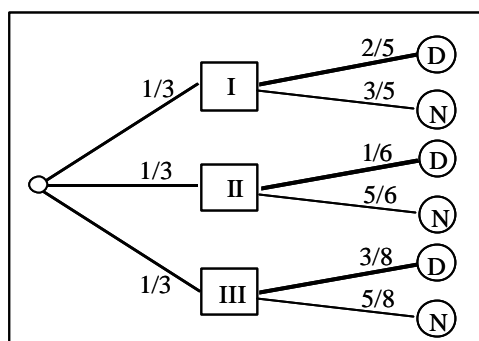


Figura 2.2: Exemplo de Diagrama de Árvore
Adaptado de [LIP 72,p.78-82]

A probabilidade P de uma lâmpada ser defeituosa é a soma das probabilidades dos três caminhos mutuamente exclusivos que conduzem até uma lâmpada defeituosa. A probabilidade de cada caminho é obtida pelo produto de cada trecho, portanto:

$$P = \frac{1}{3} \cdot \frac{2}{5} + \frac{1}{3} \cdot \frac{1}{6} + \frac{1}{3} \cdot \frac{3}{8}$$

O **Teorema de Bayes**, segundo [LAD 2000,p.23], é usado para rever crenças, sempre que novas evidências são coletadas durante um processo de raciocínio probabilístico. Isto também é expresso por [CAS 97,p.80-81] (usando termos estatísticos): o Teorema de Bayes é usado para atualizar a Distribuição de Probabilidade posterior (*à posteriori*) usando tanto as (probabilidades) *à priori* e as *likelihood* (possibilidades, verosimilhanças). Antes de ilustrar o uso deste teorema na revisão de crenças (mais adiante), será exposto agora a demonstração (bastante didática) realizada por [LIP 72,p.78-82].

Seja um espaço amostral S particionado por eventos A_1, A_2, \dots, A_n mutuamente exclusivos e seja B um evento qualquer em S (vide Figura 2.3).

$$S = (A_1 \cup A_2 \dots \cup A_n)$$

$$B = S \cap B$$

$$= (A_1 \cup A_2 \dots \cup A_n) \cap B = (A_1 \cap B) \cup (A_2 \cap B) \dots \cup (A_n \cap B)$$

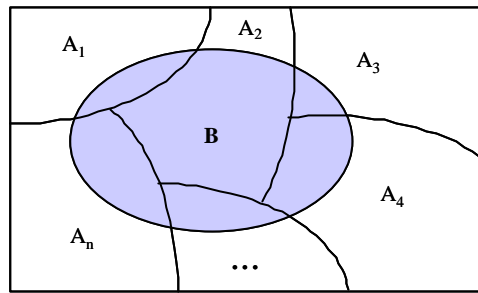


Figura 2.3: Exemplo de Partições no Espaço Amostral
Adaptado de [LIP 72,p.78-82]

Considerando que os $(A_i \cap B)$ também são mutuamente exclusivos, então

$$P(B) = P(A_1, B) + P(A_2, B) + \dots + P(A_n, B)$$

Aplicando a Regra do Produto $P(X_1, X_2) = P(X_1 | X_2).P(X_2)$ em $P(B)$, temos

$$P(B) = P(A_1).P(B | A_1) + P(A_2).P(B | A_2) + \dots + P(A_n).P(B | A_n)$$

Com relação a cada evento A_i , a probabilidade condicional de A_i dado a ocorrência de B é expressa por

$$P(A_i | B) = \frac{P(A_i, B)}{P(B)}$$

Ao substituir o denominador e aplicar a Regra do Produto no numerador, é obtido então o Teorema de Bayes

$$P(A_i | B) = \frac{P(A_i).P(B | A_i)}{P(A_1).P(B | A_1) + P(A_2).P(B | A_2) + \dots + P(A_n).P(B | A_n)}$$

Os exemplos a seguir são baseados em [LIP 72,p.78-82]. Considere uma fábrica onde as máquinas A , B e C produzem respectivamente 50%, 30% e 20% do total de peças. As taxas de produção defeituosa destas máquinas são respectivamente 3%, 4% e 5%. Portanto, qual é a probabilidade de uma peça (selecionada aleatoriamente) ser defeituosa. Se X é o evento que representa a peça defeituosa, então

$$\begin{aligned} P(X) &= P(A_1).P(X | A_1) + P(A_2).P(X | A_2) + \dots + P(A_n).P(X | A_n) \\ &= 0,5 \times 0,03 + 0,3 \times 0,04 + 0,2 \times 0,05 = 0,037 \end{aligned}$$

Este evento pode ser considerado como um processo estocástico, representado no Diagrama de Árvore da Figura 2.4.

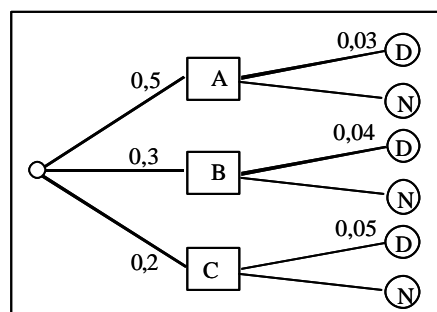


Figura 2.4: Visualização do Exemplo de Processo Estocástico
Adaptado de [LIP 72]

Agora, supondo que uma peça selecionada aleatoriamente seja defeituosa, qual é a probabilidade da máquina A ter produzido ela? Isto é, qual é $P(A|X)$? Aplicando o Teorema de Bayes

$$P(A|X) = \frac{P(A).P(X|A)}{P(A).P(X|A) + P(B).P(X|B) + P(C).P(X|C)}$$

$$= \frac{0,5 \times 0,03}{0,5 \times 0,03 + 0,3 \times 0,04 + 0,2 \times 0,05} = 0,4$$

Isto equivale à divisão da probabilidade do caminho ($\rightarrow A \rightarrow D$), pela probabilidade de todos os caminhos que conduzem à uma peça defeituosa ($\rightarrow A \rightarrow D$) + ($\rightarrow B \rightarrow D$) + ($\rightarrow C \rightarrow D$).

Retornando à aplicação do Teorema de Bayes na revisão de crenças, [CAR 99] apresenta um exemplo onde existem as hipóteses H_1 , H_2 , H_3 e H_4 com probabilidades *a priori* $P(H_i)$ apresentadas na Tabela 2.7. As crenças destas hipóteses são revistas na medida em que são consideradas as evidências E_1 e E_2 (as quais são condicionalmente independentes). As probabilidades condicionais de *likelihood* (possibilidade, verossimilhança) $P(E_1|H_i)$ e $P(E_2|H_i)$ também estão na Tabela 2.7.

Tabela 2.7: Valores das probabilidades *a priori* e condicionais

i	1	2	3	4
$P(H_i)$	0,1	0,4	0,3	0,2
$P(E_1 H_i)$	0,8	0,9	0,4	0,1
$P(E_2 H_i)$	0,7	0	0,2	0,4

Fonte: [CAR 99]

Após a observação da evidência E_1 , o cálculo das probabilidades *a posteriori* (usando o Teorema de Bayes) mostra o quanto a crença de cada hipótese aumenta ou diminui – dependendo do quanto a evidência suporta cada hipótese (vide Tabela 2.8).

Tabela 2.8: Probabilidades condicionais (*a posteriori*) de todas as hipóteses, dada a evidência E_1

I	$P(H_i E_1)$
1	$\frac{0,8 \times 0,1}{0,8 \times 0,1 + 0,9 \times 0,4 + 0,4 \times 0,3 + 0,1 \times 0,2} = 0,14$
2	$\frac{0,9 \times 0,4}{0,8 \times 0,1 + 0,9 \times 0,4 + 0,4 \times 0,3 + 0,1 \times 0,2} = 0,62$
3	$\frac{0,4 \times 0,3}{0,8 \times 0,1 + 0,9 \times 0,4 + 0,4 \times 0,3 + 0,1 \times 0,2} = 0,21$
4	$\frac{0,1 \times 0,2}{0,8 \times 0,1 + 0,9 \times 0,4 + 0,4 \times 0,3 + 0,1 \times 0,2} = 0,03$

Fonte: [CAR 99]

O mesmo processo se repete após a observação da evidência E_2 (a qual é independente de E_1), conforme exposto na Tabela 2.9.

Tabela 2.9: Probabilidades condicionais (*à posteriori*) de todas as hipóteses, dada a evidência E_2

i	$P(H_i E_1, E_2)$
1	$\frac{0,8 \times 0,7 \times 0,1}{0,8 \times 0,7 \times 0,1 + 0,9 \times 0 \times 0,4 + 0,4 \times 0,2 \times 0,3 + 0,1 \times 0,4 \times 0,2} = 0,64$
2	$\frac{0,9 \times 0 \times 0,4}{0,8 \times 0,7 \times 0,1 + 0,9 \times 0 \times 0,4 + 0,4 \times 0,2 \times 0,3 + 0,1 \times 0,4 \times 0,2} = 0$
3	$\frac{0,4 \times 0,2 \times 0,3}{0,8 \times 0,7 \times 0,1 + 0,9 \times 0 \times 0,4 + 0,4 \times 0,2 \times 0,3 + 0,1 \times 0,4 \times 0,2} = 0,27$
4	$\frac{0,1 \times 0,4 \times 0,2}{0,8 \times 0,7 \times 0,1 + 0,9 \times 0 \times 0,4 + 0,4 \times 0,2 \times 0,3 + 0,1 \times 0,4 \times 0,2} = 0,09$

Fonte: [CAR 99]

A Figura 2.5 mostra como as crenças das hipóteses foram sendo alteradas, desde as probabilidades *à priori*, passando pela avaliação do suporte da evidência E_1 , até finalmente avaliar E_2 . Note que após a observação de E_1 e E_2 , a hipótese H_2 é descartada e somente as hipóteses H_1 , H_3 e H_4 serão consideradas na medida em que novas evidências aparecerem.

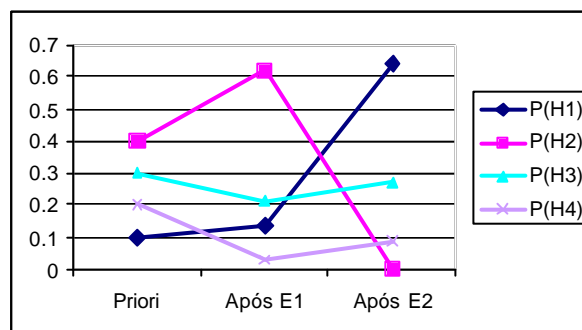


Figura 2.5: Exemplo de Atualização de Crenças (usando evidências E_i)

2.5 Redes Probabilísticas

Na subseção 2.2 foi apresentada a definição de Distribuição de Probabilidade Conjunta (DPC) para uma coleção de Variáveis Aleatórias (e a Tabela 2.5 forneceu um exemplo). Uma tabela de DPC possibilita o cálculo de Probabilidades Marginais e também de Probabilidades Condicionais, provendo desta forma as informações necessárias para as possíveis consultas sobre o domínio representado. Na prática, porém, não é fácil manter uma DPC porque seu crescimento é exponencial (aumenta na proporção do número de variáveis e número de estados destas variáveis) e além disso é uma tabela bastante esparsa, visto que nem todas as combinações de estados dos subconjuntos de variáveis possuem valores significativos de DPC.

Uma maneira de criar uma representação mais concisa de uma DPC é apresentada em [RUS 95, p.436-437]. Utiliza-se uma estrutura chamada **Rede de Crença** (também denominada como **Rede Probabilística**, Rede Bayesiana, ou Rede Causal) para representar somente as relações de dependência entre as Variáveis Aleatórias de uma DPC. A Rede Probabilística é um Grafo Acíclico Direcionado (*Direct Acyclic Graph – DAG*) que possui as seguintes características:

- 1) Os nodos da rede representam um conjunto de Variáveis Aleatórias;
- 2) Existem arcos direcionados que conectam pares de nodos, sendo que o nodo apontado pelo arco recebe influência direta do nodo que o aponta;
- 3) Cada nodo apontado possui uma tabela de probabilidade condicional que quantifica a influência/efeito do(s) nodo(s) que o apontam;

A Rede Probabilística representa então uma DPC de forma mais concisa, e é constituída de informação qualitativa (as relações de dependência, expressas na topologia da rede) e as informações quantitativas (as probabilidades e probabilidades condicionais). A Figura 2.6 mostra um exemplo de Rede Probabilística.

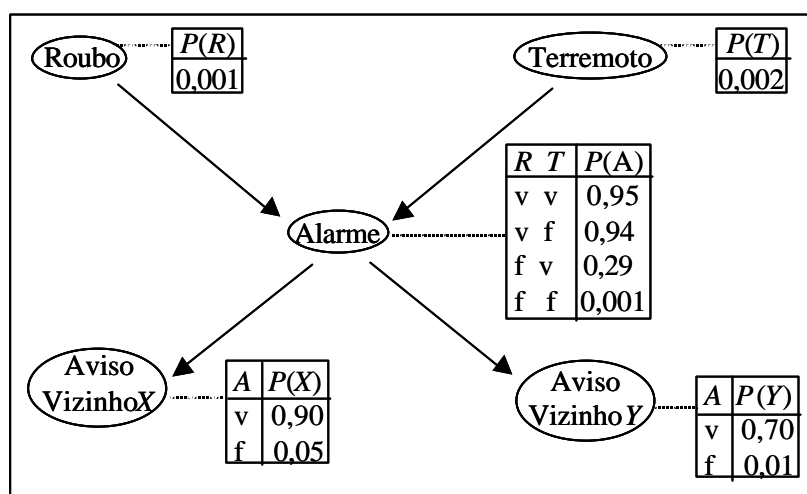


Figura 2.6: Exemplo de Rede Probabilística
Adaptação de [RUS 95, p.439]

No domínio representado pela Rede Probabilística da Figura 2.6, existem as probabilidades de roubo em residências $P(R)=0,001$ e de terremoto $P(T)=0,002$. Numa certa residência, o alarme pode disparar por causa de roubo e/ou terremoto com os valores de probabilidade condicional $P(A/R,T)$ apresentados na tabela ligada à *Alarme*. Além disso, o dono da residência (do alarme) combinou com seus dois vizinhos X e Y, que estes o avisassem a cada vez que eles acharem que o alarme disparou. O vizinho X, quando escuta o alarme, sempre avisa corretamente com probabilidade $P(X/A=v)=0,90$. Mas às vezes ele confunde o som do alarme com outros ruídos e avisa incorretamente assim mesmo, com probabilidade $P(X/A=f)=0,05$. Por outras razões, o vizinho Y possui $P(Y/A=v)=0,70$ e $P(X/A=f)=0,01$. É importante observar que a Rede Probabilística resume ou sumariza várias situações possíveis, as quais não são modeladas explicitamente. Por exemplo, a possibilidade do alarme disparar por outro motivo além de roubo ou terremoto (defeito técnico, ação de algum inseto, etc) está resumida na probabilidade $P(A/R=f,T=f)=0,001$.

A semântica de uma Rede Probabilística, segundo [RUS 95, p.438], pode ser entendida de duas maneiras: Como uma representação de uma Distribuição de Probabilidade Conjunta (DPC); ou, como uma codificação de uma coleção de declarações de independências condicionais.

Considerando a semântica da representação de uma DPC, cada entrada genérica de uma DPC, $P(V_{1=v_1}, \dots, V_{n=v_n})$ é representada na Rede Probabilística como o produto dos elementos das tabelas de probabilidade e de probabilidade condicionais, cujo valor é obtido pela fórmula:

$$P(V_1 = v_1, \dots, V_2 = v_2) = \prod_{i=1}^n P(v_i | \text{Pais}(V_i))$$

onde $\text{Pais}(V_i)$ são os nodos que causam, isto é, possuem arcos que apontam para V_i .

Para exemplificar o cálculo deste produto, considere ainda a Figura 2.6. Deseja-se saber a probabilidade de ambos os vizinhos X e Y avisarem corretamente, dado que o alarme A disparou de fato, mas não por causa de roubo R ou terremoto T .

$$\begin{aligned} P(X_{=v}, Y_{=v}, A_{=v}, R_{=f}, T_{=f}) &= P(X_{=v}|A_{=v}) \cdot P(Y_{=v}|A_{=v}) \cdot P(A_{=v}|R_{=f}, T_{=f}) \cdot P(R_{=f}) \cdot P(T_{=f}) \\ &= 0,90 \times 0,70 \times 0,001 \times 0,999 \times 0,998 = 0,00062 \end{aligned}$$

Um exemplo mais completo é apresentado agora para ilustrar como uma Rede Probabilística representa de forma mais compacta uma Distribuição de Probabilidade Conjunta. A Figura 2.7-A mostra uma pequena Rede Probabilística com suas respectivas tabelas de probabilidades, sendo que a sua Distribuição de Probabilidade Conjunta (DPC) equivalente é mostrada na Figura 2.7-B.

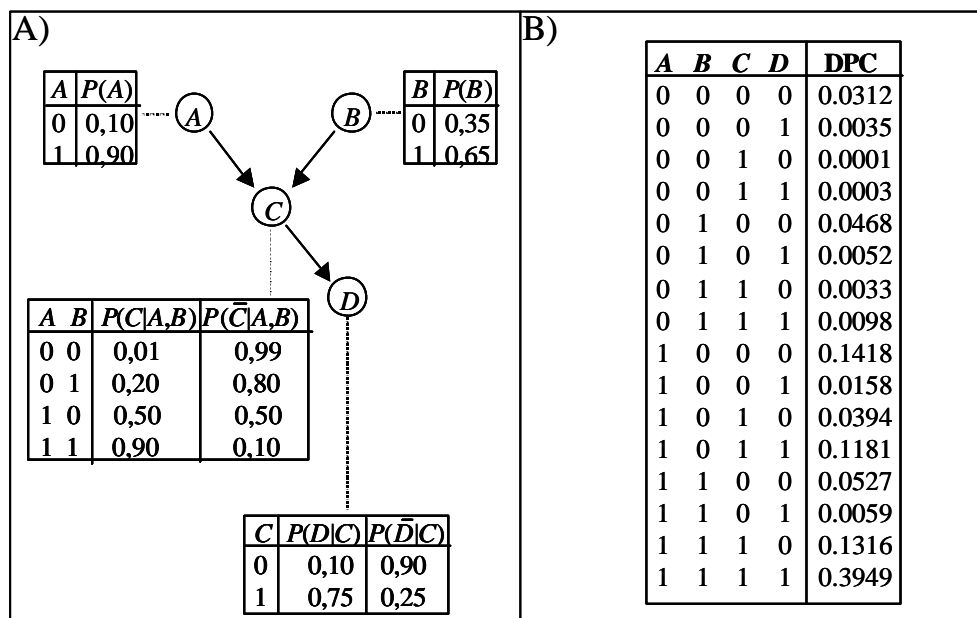


Figura 2.7: Rede Probabilística e sua DPC equivalente

Na Figura 2.7-A, as Probabilidades Condicionais $P(\bar{C}|A,B)$ e $P(\bar{D}|C)$ são mostradas apenas para facilitar a exposição do assunto, pois estes dados são os complementos obtidos das Probabilidades Condicionais $P(C|A,B)$ e $P(D|C)$ já representadas na rede.

Vamos mostrar agora como a Rede Probabilística representa a DPC, ou seja $P(A,B,C,D)$. Uma vez que a ordem das Variáveis Aleatórias não influi no valor da DPC, elas ficarão na ordem $P(D,C,B,A)$, para que a aplicação da Regra da Cadeia (mostrada na seção 2.4) gere a expressão mais próxima da rede representada:

$$P(D,C,B,A) = P(D|C,B,A) \cdot P(C|B,A) \cdot P(B|A) \cdot P(A)$$

A topologia da rede informa que D é condicionalmente dependente de C , portanto $P(D|C,B,A)$ se reduz a $P(D|C)$, e que B é condicionalmente independente de A , portanto $P(B|A)$ se reduz a $P(B)$. Então a DPC pode ser assim expressa:

$$P(A,B,C,D) = P(A) \cdot P(B) \cdot P(C|B,A) \cdot P(D|C)$$

Cada valor de DPC pode ser construído com as informações da Rede Probabilística, por exemplo:

$$\begin{aligned}
 P(A_{=0}, B_{=0}, C_{=0}, D_{=0}) &= P(A_{=0}) \cdot P(B_{=0}) \cdot P(C_{=0}/B_{=0}, A_{=0}) \cdot P(D_{=0}/C_{=0}) \\
 &= 0,10 \cdot 0,35 \cdot 0,99 \cdot 0,90 = 0,0312
 \end{aligned}$$

Este cálculo é repetido para todas as variações de estados das variáveis, até a última:

$$\begin{aligned}
 P(A_{=1}, B_{=1}, C_{=1}, D_{=1}) &= P(A_{=1}) \cdot P(B_{=1}) \cdot P(C_{=1}/B_{=1}, A_{=1}) \cdot P(D_{=1}/C_{=1}) \\
 &= 0,90 \cdot 0,65 \cdot 0,90 \cdot 0,75 = 0,3949
 \end{aligned}$$

Desta forma, a Rede Probabilística usa os relacionamentos de dependência condicional para reduzir o tamanho da Tabela de DPC.

Considerando novamente aquela semântica onde a Rede Probabilística codifica uma coleção de declarações de independências condicionais, [RUS 95, p.438] afirma que esta semântica é útil para o projeto de algoritmos de inferência (probabilística). Em geral, dado uma Rede Probabilística, deseja-se saber que conjunto de nodos X é independente de outro conjunto Y , após o conhecimento dos estados de alguns nodos E (isto é, o conjunto dos nodos de Evidência). A notação usada nesta declaração é $I(X, E, Y)$.

Antes de expor as regras que orientam a construção das declarações de independência condicional, é necessário apresentar (baseando-se em [JEN 01, p.6-11]) os conceitos de conexão serial, conexão divergente e conexão convergente.

- **Conexão Serial.** A Figura 2.8 mostra uma Conexão Serial, onde A influencia/causa B , e este influencia C .

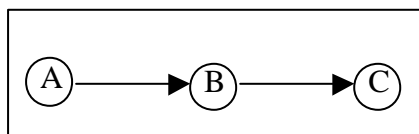


Figura 2.8: Conexão serial. Quando B é instanciado, ele bloqueia a comunicação entre A e C

Fonte: [JEN 2001, p.6]

Quando o valor de A é conhecido/instanciado (ou seja, quando A é evidência), A influencia a certeza de B , que por sua vez influencia a certeza de C . A mesma propagação de influência ocorre, porém em sentido contrário, quando C é a evidência.

Todavia, se B é evidência, então ele bloqueia a comunicação entre A e C , os quais se tornam condicionalmente independentes – dado B , isto é $I(A, B, C)$.

- **Conexão Divergente.** A Figura 2.9 mostra uma Conexão Divergente, onde A influencia/causa todos os nodos “filhos” $B, C \dots E$.

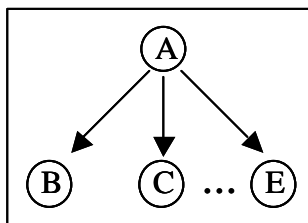


Figura 2.9: Conexão Divergente. Se A é instanciado, ele bloqueia a comunicação entre seus “filhos”

Fonte [JEN 2001, p.7]

Se A não for evidência, então os nodos “filhos” podem influenciar uns aos outros. Mas se A for evidência, então ele bloqueia a comunicação entre seus “filhos”, isto é, eles se tornam condicionalmente independentes – dado A , por exemplo $I(B,A,C)$.

Para ilustrar o conceito de Conexão Divergente, [JEN 2001,p.7] usa o exemplo da Figura 2.10, onde aparecem as relações causais entre *Sexo* (*Masculino*,*Feminino*), *Comprimento_de_Cabelo* (*longo*, *curto*), e *Estatura* (<168 cm, >168 cm).

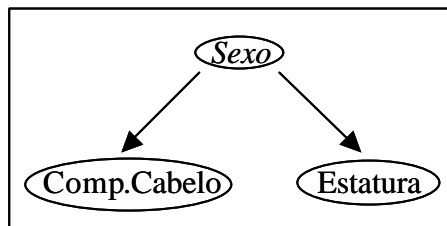


Figura 2.10: A variável *Sexo* influencia as variáveis *Comp.Cabelo* e *Estatura*

Fonte: [JEN 2001,p.7]

Quando não se sabe o sexo de alguém (a variável *Sexo* não foi instanciada), o conhecimento sobre *Comprimento_de_Cabelo* é um indicador a respeito da variável *Sexo* – que por sua vez influencia o foco da crença sobre *Estatura*. Por outro lado, se *Sexo*=*Masculino* então o conhecimento sobre *Comprimento_de_Cabelo* não acrescenta nenhum indício sobre *Estatura*.

- **Conexão Convergente.** A Figura 2.11 mostra uma Conexão Convergente. Neste caso, se A não é evidência (se não foi instanciado), então os nodos “pais” (B , C ... E) são condicionalmente independentes. Ou seja, o conhecimento sobre uma possível causa de um evento (p.ex. B) não influencia em nada na certeza sobre outras possíveis causas (p.ex. C ou ... E). Todavia, se A é evidência (isto é, se algo é conhecido sobre as conseqüências), então uma informação sobre uma possível causa (p.ex. B) influencia na certeza de outras causas (p.ex. C ou ... E).

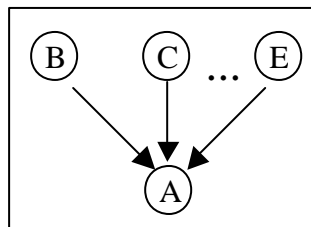


Figura 2.11: Conexão Convergente. Se A for evidência, ela abre a comunicação entre seus “pais”

Fonte: [JEN 2001,p.7]

Pode-se dizer que se A ocorreu (é evidência), então tanto B quanto C (ou ... E) podem ter causado A . Mas se B também ocorreu (causando A), então diminui a certeza de que C (ou ... E) também ocorreram. Por outro lado, se A ocorreu e é conhecido que nem C (ou ... E) ocorreram, então aumenta a

certeza de que B ocorreu. Resumindo: se A (ou algum de seus nodos “descendentes”) for evidência, então A abre a comunicação entre seus nodos “pais”.

A Conexão Convergente é ilustrada por [JEN 2001,p.8] através de um exemplo sobre diagnóstico médico. A Figura 2.12 mostra as relações causais entre *Infeção* (*Salmonella*), *Gripe*, *Náusea* e *Palidez*.

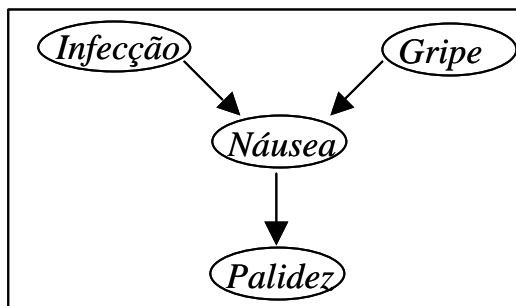


Figura 2.12: *Infeção* (*Salmonella*) e *Gripe* podem causar *Náusea*, que por sua vez causa *Palidez*

Fonte: [JEN 2001,p.8]

Se não existe informação sobre *Náusea* nem *Palidez*, a informação de um paciente com *Infeção* não diz nada a respeito de *Gripe*. Todavia, se é conhecido que um paciente apresenta *Palidez*, então a informação de que ele não possui *Infeção* aumentam as chances dele estar com *Gripe*.

Após apresentar os conceitos das Conexões Serial, Divergente e Convergente, podemos considerar novamente aquela semântica onde a Rede Probabilística codifica uma coleção de declarações de independências condicionais. A importância desta semântica é que ela serve de base para o projeto de algoritmos de inferência probabilística [RUS 95, p.438].

A questão fundamental é saber, dado uma Rede Probabilística, que conjunto de nodos (ou variáveis) X é independente de outro conjunto Y , após o conhecimento dos estados de alguns nodos E (isto é, o conjunto dos nodos de Evidência). A resposta para esta questão é provida pela definição da **Separação Dependente de Direção** (*Direction-Dependent Separation*) ou simplesmente, **d-Separação**. A definição formal e as respectivas provas das propriedades estão em [PEA 88,p.117-122]. A definição da d-Separação a ser apresentada agora é baseada em [JEN 2001,p.10]. Duas variáveis distintas X e Y numa Rede Probabilística estão d-Separadas se, para todos os caminhos da conexão entre X e Y , existe uma variável intermediária Z (distinta de X e Y) tal que:

- 1) A conexão é serial e Z é instanciada (Z pertence ao conjunto E);
- 2) A conexão é divergente e Z é instanciada ($Z \in E$);
- 3) A conexão é convergente, e nem Z ou nenhum dos “descendentes” Z' de Z foram instanciados ($Z \notin E$ ou $Z' \notin E$).

Quando X e Y não estão d-Separados, é dito então que eles estão d-Conectados. A Figura 2.13 ilustra estes três casos.

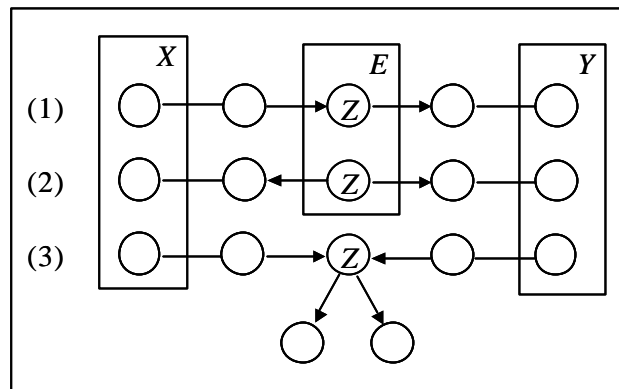


Figura 2.13: Os Três Casos de d-Separação numa Rede Probabilística
Fonte: [RUS 95,p.445]

2.6 Tipos Básicos de Inferência Probabilística

Em uma Rede Probabilística, existem duas operações básicas: Revisão de Crenças e Atualização de Crenças (também conhecida como Inferência Probabilística). Segundo [LAD 2000,p.34], a Revisão de Crenças é usada para construir explicações ou diagnósticos, através da aceitação de um subconjunto de hipóteses (que consideradas ao mesmo tempo) constituem a explicação mais satisfatória da evidência disponível. “Em termos probabilísticos, equivale a encontrar a instanciação mais provável de todas as variáveis das hipóteses, face aos dados observados. O resultado é uma lista ótima de proposições aceitas em conjunto, lista que pode mudar abruptamente quando mais evidência é obtida.” [PEA 88,p.240 Apud LAD 2000,p.34].

Por sua vez, a Atualização de Crenças (isto é, a **Inferência Probabilística**) “refere-se à fusão e propagação do impacto de novas evidências e crenças, através da rede bayesiana (isto é, a Rede Probabilística), de forma tal que a cada proposição seja atribuída uma medida de certeza compatível com os axiomas da teoria da probabilidade” [PEA 88,p.143 Apud LAD 2000,p.35]. Sobre um sistema de Inferência Probabilística, [RUS 95,p.445] afirma que sua tarefa básica é calcular a Distribuição de Probabilidade posterior (*à posteriori*) para um conjunto de variáveis de consulta, uma vez observados valores exatos para algumas variáveis de evidência. Na notação apropriada, isto equivale à $P(\text{Consulta}|\text{Evidência})$.

A Inferência Probabilística pode ser feita de três tipos básicos [RUS 95,p.447]: **Inferência Causal** (ou *top-down*), **Inferência de Diagnóstico** (ou *bottom-up*) e **Inferência Intercausal** (ou *explaining away*). A Figura 2.14 ilustra estas inferências básicas, onde estão representados os nodos de Consulta (*C*) e de Evidência (*E*).

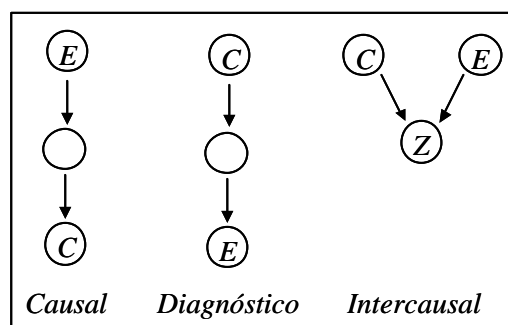


Figura 2.14: Exemplos dos tipos básicos de Inferência Probabilística
Adaptação de [RUS 95,p.448]

Para ilustrar os conceitos destes três tipos básicos de Inferência Probabilística, é usada a rede da Figura 2.12, onde *Infecção* (I) e *Gripe* (G) causam *Náusea* (N), que por sua vez causa *Palidez* (P). Ou seja,

$$\begin{array}{l} \text{Infecção} \rightarrow \\ \text{Gripe} \rightarrow \end{array} \text{Náusea} \rightarrow \text{Palidez}$$

No entanto os valores das probabilidades são oriundos da Figura 2.7, (visto que as duas redes possuem a mesma topologia, isto é, o mesmo número de variáveis e as mesmas relações causais). Ou seja, as relações causais (da Figura 2.12) mostradas acima correspondem às relações causais (da Figura 2.7) mostradas abaixo:

$$\begin{array}{l} A \rightarrow \\ B \rightarrow \end{array} C \rightarrow D$$

2.6.1 Inferência Causal (ou *top-down*)

Dado que *Gripe* (G) é evidência, isto é, foi instanciado o valor $G_{=1}$, então se deseja saber qual é a probabilidade de *Palidez* (P) uma vez observada a *Gripe*, ou seja $P(P_{=1}|G_{=1})$. Esta inferência é Causal porque *Gripe* (G) - juntamente com *Infecção* (I) - causam *Náusea* (N), que por sua vez causa *Palidez* (P).

Considerando a topologia da rede, *Palidez* é condicionalmente dependente de *Náusea* (que não é evidência). Então, $P(P_{=1}|G_{=1})$ é reescrita para incluir *Náusea*, considerando todos os estados possíveis de N (isto é representado por Σ_N).

$$P(P_{=1} | G_{=1}) = \sum_N P(P_{=1}, N | G_{=1})$$

A probabilidade $P(P_{=1}, N | G_{=1})$ é reescrita usando o conceito de Probabilidade Condicional.

$$P(P_{=1}, N | G_{=1}) = \frac{P(P_{=1}, N, G_{=1})}{P(G_{=1})}$$

O numerador $P(P_{=1}, N, G_{=1})$ é reescrito usando a Regra da Cadeia.

$$P(P_{=1}, N | G_{=1}) = \frac{P(P_{=1} | N, G_{=1}) \cdot P(N | G_{=1}) \cdot P(G_{=1})}{P(G_{=1})}$$

Cortando $P(G_{=1})$ no numerador e no denominador, e substituindo no Σ_N .

$$P(P_{=1} | G_{=1}) = \sum_N P(P_{=1} | N, G_{=1}) \cdot P(N | G_{=1})$$

A topologia da rede informa que *Palidez* e *Gripe* são condicionalmente independentes, dado *Náusea*. Portanto, $P(P_{=1}|N, G_{=1})$ se reduz a $P(P_{=1}|N)$. Com relação a $P(N|G_{=1})$, a topologia também informa que *Náusea* é condicionalmente dependente de *Gripe* (que é evidência: $G_{=1}$) mas também de *Infecção* (que não é evidência). Sendo assim, $P(N|G_{=1})$ é reescrita para incluir I : $P(N, I|G_{=1})$, considerando todos os estados possíveis de I (isto é representado por Σ_I).

$$P(P_{=1} | G_{=1}) = \sum_N P(P_{=1} | N) \cdot \left[\sum_I P(N, I | G_{=1}) \right]$$

Em $P(N, I | G_{=1})$ também são aplicadas transformações e otimizações para reescrevê-la como $P(N | I, G_{=1}) \cdot P(I | G_{=1})$ e como *Infeção* é condicionalmente independente de *Gripe*, $P(I | G_{=1})$ se reduz a $P(I)$, como segue.

$$P(P_{=1} | G_{=1}) = \sum_N P(P_{=1} | N) \cdot \left[\sum_I P(N | I, G_{=1}) \cdot P(I) \right]$$

A Figura 2.15 mostra o Diagrama de árvore desta expressão. Uma vez observada a evidência de *Gripe* $G_{=1}$, e fixado o valor $P_{=1}$ da probabilidade de *Palidez* que se deseja descobrir o valor $P(P_{=1} | G_{=1})$, o diagrama em questão mostra todos os caminhos possíveis que ligam $P_{=1}$ e $G_{=1}$ - considerando todos os estados possíveis das variáveis *Náusea* (intermediária entre *Palidez* e *Gripe*) e *Infeção* (relacionada com *Náusea*, juntamente com *Gripe*, pois *Infeção* e *Gripe* causam *Náusea*).

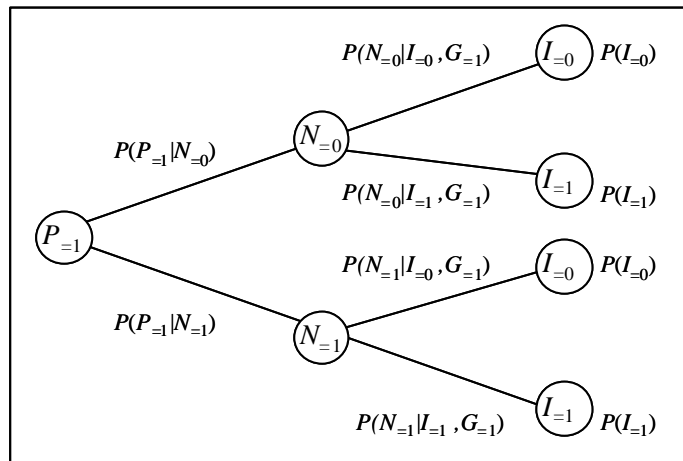


Figura 2.15: Diagrama de Árvore do Exemplo de Inferência Causal

O valor de $P(P_{=1} | G_{=1})$ é obtido somando a probabilidade de cada caminho (sendo que cada caminho é o produto da probabilidade de cada trecho). A área em cor cinza indica que os valores de $P_{=1}$ e $G_{=1}$ estão fixados.

$$\begin{aligned} P(P_{=1} | G_{=1}) &= P(P_{=1} | N_{=0}) \cdot P(N_{=0} | I_{=0}, G_{=1}) \cdot P(I_{=0}) + \\ &P(P_{=1} | N_{=0}) \cdot P(N_{=0} | I_{=1}, G_{=1}) \cdot P(I_{=1}) + \\ &P(P_{=1} | N_{=1}) \cdot P(N_{=1} | I_{=0}, G_{=1}) \cdot P(I_{=0}) + \\ &P(P_{=1} | N_{=1}) \cdot P(N_{=1} | I_{=1}, G_{=1}) \cdot P(I_{=1}) \end{aligned}$$

Substituindo pelos valores das tabelas de probabilidades da Figura 2.7-A, temos:

$$\begin{aligned} P(P_{=1} | G_{=1}) &= 0,10 \times 0,80 \times 0,10 + \\ &0,10 \times 0,10 \times 0,90 + \\ &0,75 \times 0,20 \times 0,10 + \\ &0,75 \times 0,90 \times 0,90 \\ &= 0,6395 \end{aligned}$$

Este mesmo valor é obtido se forem usados os valores de DPC da Figura 2.7-B, calculando $P(P_{=1} | G_{=1})$ usando a definição de Probabilidade Condicional, e depois gerando $P(P_{=1}, G_{=1})$ e $P(G_{=1})$ através do cálculo de Probabilidade Marginal.

$$P(P_{=1} | G_{=1}) = \frac{P(P_{=1}, G_{=1})}{P(G_{=1})} = \frac{0,4157}{0,6500} = 0,6395$$

O problema é que, conforme já dito anteriormente, manter uma tabela de DPC é muito difícil na prática, onde o domínio da aplicação envolve muitas Variáveis Aleatórias.

2.6.2 Inferência de Diagnóstico (ou *bottom-up*)

Ainda usando a Rede Probabilística da Figura 2.12 (com as probabilidades da Figura 2.7), ou seja, ainda usando as seguintes relações causais

$$\begin{array}{l} \text{Infecção} \rightarrow \\ \text{Gripe} \rightarrow \end{array} \begin{array}{l} \text{Náusea} \rightarrow \\ \text{Palidez} \end{array}$$

Considere agora a situação inversa do exemplo anterior (sobre Inferência Causal). Agora, *Palidez* $P_{=1}$ é a evidência e então deseja-se saber a probabilidade de *Gripe* uma vez observada a *Palidez*, ou seja $P(G_{=1}|P_{=1})$. Esta inferência é de Diagnóstico porque *Palidez* é efeito de *Náusea*, que por sua vez é efeito de *Gripe* (que, juntamente com *Infecção*, causa *Náusea*).

O procedimento básico consiste em reescrever $P(G_{=1}|P_{=1})$ de maneira a serem aplicadas as técnicas mostradas anteriormente para Inferência Causal. Para isto, [NIL 98,p.329] aponta o uso da Regra de Bayes.

$$P(G_{=1} | P_{=1}) = \frac{P(P_{=1} | G_{=1}) \cdot P(G_{=1})}{P(P_{=1})}$$

No numerador, a probabilidade $P(P_{=1}|G_{=1})= 0,6395$ foi calculada no exemplo anterior. O valor de $P(G_{=1})=0,65$ já está disponível na rede (a variável *B* na Figura 2.7). Resta agora calcular $P(P_{=1})$ no denominador, sendo que existem dois modos. O primeiro é mais direto, fazendo algumas reescritas

$$\begin{aligned} P(P_{=1}) &= P(P_{=1}, G_{=0}) + P(P_{=1}, G_{=1}) \\ &= P(P_{=1}|G_{=0}) \cdot P(G_{=0}) + P(P_{=1}|G_{=1}) \cdot P(G_{=1}) \end{aligned}$$

Ou, numa visão alternativa: $P(G_{=0}|P_{=1}) + P(G_{=1}|P_{=1}) = 1$
Então:

$$\frac{P(P_{=1} | G_{=0}) \cdot P(G_{=0})}{P(P_{=1})} + \frac{P(P_{=1} | G_{=1}) \cdot P(G_{=1})}{P(P_{=1})} = 1$$

Ao substituir a reescrita de $P(P_{=1})$ na primeira expressão, obtém-se a aplicação do Teorema de Bayes:

$$P(G_{=1} | P_{=1}) = \frac{P(P_{=1} | G_{=1}) \cdot P(G_{=1})}{P(P_{=1} | G_{=0}) \cdot P(G_{=0}) + P(P_{=1} | G_{=1}) \cdot P(G_{=1})}$$

Nesta última expressão, só resta calcular $P(P_{=1}|G_{=0})$ no denominador, usando as técnicas de Inferência Causal. O resultado é $P(P_{=1}|G_{=0}) = 0,3932$. Finalmente, substitui-se todos os valores em $P(P_{=1}|G_{=1})$

$$P(G_{=1} | P_{=1}) = \frac{0,6395 \times 0,65}{0,3932 \times 0,35 + 0,6395 \times 0,65} = 0,7513$$

Retornando aos modos de calcular $P(P_{=1})$, o primeiro modo é mais direto explorando a propriedade de Complemento de Probabilidade: $P(G_{=0}/P_{=1})+P(G_{=1}/P_{=1})=1$. Já o segundo modo, é menos direto, e será apresentado apenas para efeito didático. Trata-se de usar a expressão da DPC para a Rede Probabilística da Figura 2.7-A:

$$P(A,B,C,D) = P(A).P(B).P(C/A,B).P(D/C)$$

Cuja tradução para as variáveis da rede da Figura 2.12 corresponde a

$$P(I,G,N,P) = P(I).P(G).P(N/I,G).P(P/N)$$

Para depois calcular a Probabilidade Marginal de $P(P_{=1})$, isto é, somando todas as DPC variando todos os estados das demais variáveis.

$$P(P_{=1}) = \sum_{I,G,N} P(P_{=1} | N).P(N | I, G).P(I).P(G)$$

O Teorema da Multiplicação permite reescrever este somatório de uma maneira mais intuitiva, para apresentar o diagrama de árvore na Figura 2.16.

$$P(P_{=1}) = \sum_N P(P_{=1} | N). \sum_{I,G} P(N | I, G).P(I).P(G)$$

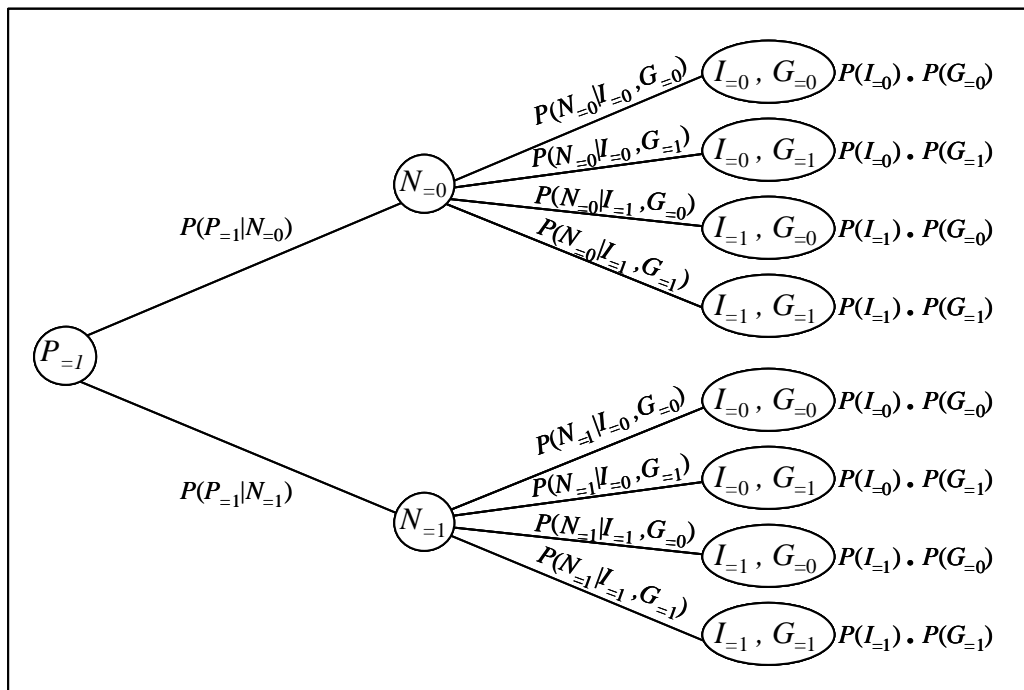


Figura 2.16: Diagrama de Árvore do Cálculo da Probabilidade Marginal

O valor de $P(P_{=1})$ é obtido somando a probabilidade de cada caminho (sendo que cada caminho é o produto da probabilidade de cada trecho). A área em cor cinza indica que o valor de $P_{=1}$ está fixado.

$$\begin{aligned}
P(P_{=1}) = & P(P_{=1}|N_{=0}). P(N_{=0}|I_{=0}, G_{=0}). P(I_{=0}). P(G_{=0}) + \\
& P(P_{=1}|N_{=0}). P(N_{=0}|I_{=0}, G_{=1}). P(I_{=0}). P(G_{=1}) + \\
& P(P_{=1}|N_{=0}). P(N_{=0}|I_{=1}, G_{=0}). P(I_{=1}). P(G_{=0}) + \\
& P(P_{=1}|N_{=0}). P(N_{=0}|I_{=1}, G_{=1}). P(I_{=1}). P(G_{=1}) + \\
& P(P_{=1}|N_{=1}). P(N_{=1}|I_{=0}, G_{=0}). P(I_{=0}). P(G_{=0}) + \\
& P(P_{=1}|N_{=1}). P(N_{=1}|I_{=0}, G_{=1}). P(I_{=0}). P(G_{=1}) + \\
& P(P_{=1}|N_{=1}). P(N_{=1}|I_{=1}, G_{=0}). P(I_{=1}). P(G_{=0}) + \\
& P(P_{=1}|N_{=1}). P(N_{=1}|I_{=1}, G_{=1}). P(I_{=1}). P(G_{=1})
\end{aligned}$$

2.6.3 Inferência Intercausal (ou Explaining Away)

Mais uma vez, é usada a Rede Probabilística da Figura 2.12 (com as probabilidades da Figura 2.7), ou seja, ainda usando as seguintes relações causais

$$\begin{array}{l}
\text{Infecção} \rightarrow \\
\text{Gripe} \rightarrow
\end{array}
\begin{array}{l}
\text{Náusea} \rightarrow \\
\text{Palidez}
\end{array}$$

Para explicar a Inferência Intercausal, é repetido o texto associado ao exemplo da Figura 2.12 - para depois acrescentar uma análise sobre os cálculos das probabilidades: “Se não existe informação sobre *Náusea* nem *Palidez*, a informação de um paciente com *Infecção* não diz nada a respeito de *Gripe*. Todavia, se é conhecido que um paciente apresenta *Palidez*, então a informação de que ele não possui *Infecção* aumentam as chances dele estar com *Gripe*.”

Primeiramente, consideremos o *trecho1*: “Se não existe informação sobre *Náusea* nem *Palidez*, a informação de um paciente com *Infecção* não diz nada a respeito de *Gripe*.”

- A probabilidade (*à priori*) de *Gripe* é $P(G_{=1})=0,65$.
- O *trecho1* pode ser reescrito numa expressão de probabilidade condicional, cujo valor é $P(G_{=1}|I_{=1})=0,65$. Ou seja, como *Gripe* e *Infecção* são condicionalmente independentes, $P(G_{=1})=P(G_{=1}|I_{=1})$.

Agora, considerando o *trecho2*: “Todavia, se é conhecido que um paciente apresenta *Palidez*, então a informação de que ele não possui *Infecção* aumentam as chances dele estar com *Gripe*.”

- O *trecho2* pode ser reescrito numa expressão de probabilidade condicional, cujo valor é $P(G_{=1}|I_{=0}, P_{=1})=0,8004$. Ou seja, a probabilidade de *Gripe* aumentou de $P(G_{=1})=0,65$ (*à priori*) para $P(G_{=1}|I_{=0}, P_{=1})=0,8004$ (*à posteriori*, dado $I_{=0}$ e $P_{=1}$).
- Mas, se o *trecho2* for alterado para indicar a ocorrência de *Palidez* e *Infecção*, então a probabilidade de ter *Gripe* é $P(G_{=1}|I_{=1}, P_{=1})=0,7496$. Ou seja, a probabilidade de *Gripe* aumentou de $P(G_{=1})=0,65$ para $P(G_{=1}|I_{=1}, P_{=1})=0,7496$. No entanto $P(G_{=1}|I_{=1}, P_{=1})$ é menor do que $P(G_{=1}|I_{=0}, P_{=1})=0,8004$ (quando *Infecção* não foi observada).

Pode ser observado que a Inferência Intercausal envolve uma Conexão Convergente (definida na seção 2.5), ou seja *Infecção* e *Gripe* convergem para (isto é, causam) *Náusea*, que por sua vez causa *Palidez*. O exemplo recém-apresentado confirma [JEN 2001,p.8]: “evidência pode apenas ser transmitida através de uma conexão convergente se tanto a variável da conexão ou uma de suas descendentes

receberam evidência”. Neste caso, a variável da conexão é *Náusea*. Ainda sobre a Inferência Intercausal, [NIL 98,p.329] comenta que esta inferência usa a técnica de Inferência Causal embutida na técnica de Inferência de Diagnóstico, sendo que o uso da Regra de Bayes é um passo importante neste processo.

2.7 Analogia da Rede Probabilística com o Modelo Relacional

O que foi apresentado até agora sobre Rede Probabilística (RP) possibilita fazer uma analogia com o Modelo de Banco de Dados Relacional (MBDR). Trabalhos como [WON 2002,WON 2001] mostram as relações que existem entre MBDR e RP. Aqui, iniciamos a analogia entre MBDR e RP ao comparar a tabela de Distribuição de Probabilidades Conjunta (DPC) com um Banco de Dados que possui uma única tabela, isto é um único esquema que reúne todos os atributos do domínio modelado. Na Figura 2.7-B foi exemplificada uma tabela de DPC. O fato comum (para RP e MBDR) é que manter uma única tabela tem suas desvantagens.

- Na caso da DPC, seu crescimento aumenta exponencialmente em função do número de variáveis e número de estados destas variáveis. Além disso, é uma tabela bastante esparsa, visto que nem todas as combinações de estados dos subconjuntos de variáveis possuem valores significativos de DPC.
- No caso do MBDR, a tabela única apresenta o problema de repetição de informação (além de se tornar também bastante esparsa). Sendo que o objetivo de um projeto de banco de dados relacional, segundo [KOR 89,p.187], é guardar informação sem redundância desnecessária, evitando as propriedades indesejáveis tal como: repetição de informação, incapacidade de representar parte da informação, e perda de informação.

Um dos objetivos comuns (entre RP e MBDR), é representar a informação do domínio de forma compacta, explorando a propriedade (apresentada em [RUS 95,p.441]) de **Sistemas Localmente Estruturados** (ou Sistemas Esparços), onde “cada subcomponente interage diretamente somente com um número de outros componentes que fazem fronteira, independente do número total de componentes”.

- A Rede Probabilística representa uma DPC modelando os relacionamentos causais entre as Variáveis Aleatórias do domínio, conforme descrito e exemplificado na seção 2.5 (vide Figura 2.7-A). “Se a rede é construída ligando variáveis influenciadas causalmente pelas anteriores então o modelo probabilístico reflete corretamente as relações de independência condicional do domínio (hipótese causal de Markov)” [LAD 2000,p.34].
- A estrutura lógica de um MBDR é representada através de um modelo conceitual que identifica um grupo básico de objetos chamados Entidades e também identifica os Relacionamentos entre estes objetos [KOR 89,p.23]. Todavia, num MBDR, a semântica do relacionamento inclui a causalidade mas não se limita a esta (por exemplo, pode ser uma semântica do tipo “Entidade A faz-parte-de Entidade B”), enquanto que a semântica de um relacionamento na Rede Probabilística é unicamente baseada na causalidade, sendo que [PEA 2000,p.25] afirma que “relacionamentos causais são ontológicos, descrevendo restrições físicas objetivas do nosso mundo”.

Tanto para RP quanto MBDR, é fundamental que a representação compacta do domínio ocorra sem perda de informação, para isto é preciso obedecer às restrições de dependência que preservam os relacionamentos existentes no domínio modelado.

- Para uma RP, os relacionamentos de independência condicional orientam a construção da topologia da rede, ao otimizar a expressão de probabilidade condicional que representa a DPC (no exemplo da Figura 2.7):

$$P(D,C,B,A) = P(D/C,B,A).P(C/B,A).P(B/A).P(A)$$

foi otimizada pelas restrições (aqui, declarações de independência condicional):

$$P(D,C,B,A) = P(D/C).P(C/B,A).P(B).P(A)$$

- Para um MBDR, a decomposição (sem perda de informação) de um esquema num subconjunto de esquemas é feita obedecendo restrições sobre dependência de dados, gerando esquemas que seguem Formas Normais. Em [KOR 89,cap.6] são apresentadas três tipos básicos de dependências de dados: Funcionais (são “referidas como dependências geradoras de igualdade” [Ibid, p.217]), Multivaloradas (são “referidas como dependências geradoras de tuplas” [Ibid, p.217]), e Dependências de Junção (“usadas para restringir - o conjunto de relações legais sobre um esquema - àquelas relações para as quais uma decomposição dada é uma decomposição de junção sem perdas” [Ibid, p.226]).

Uma vez que os relacionamentos do domínio estão representados de uma maneira compacta e sem perda de informação, estes relacionamentos podem ser reconstruídos, sendo que otimizações podem ser eventualmente aplicadas durante o processo de reconstrução dos relacionamentos.

- Para uma RP, nos procedimentos básicos de Inferência Probabilística, os caminhos dos relacionamentos causais (entre as variáveis envolvidas numa consulta) são reconstruídos durante o processo de propagação do impacto das evidências observadas, conforme ilustrado no diagrama da Figura 2.15.

Durante esta reconstrução, são consideradas todas as variações de estados das variáveis intermediárias (por exemplo, N na Figura 2.15) e também as variáveis relacionadas nas relações causais (por exemplo, I na Figura 2.15).

Também durante esta reconstrução, os critérios de d-Separação (seção 2.5) orientam as otimizações que podem ocorrer quando as variáveis intermediárias são incluídas - por exemplo, na Figura 2.15, $P(P_{=1}|N)$ é a otimização de $P(P_{=1}|N, G_{=1})$.

Outro exemplo de otimização ocorre explorando a estrutura local de uma DPC fatorada, por exemplo (proveniente de [CAS 97,p.321]), considere o cálculo da Probabilidade Marginal de uma variável D :

$$P(D) = \sum_{A,B,C,E,F,G} p(A).p(B).P(C|A).P(D|A,B).P(E).P(F|D).P(G|D,E)$$

Tendo cada variável dois estados, então o somatório realiza $2^6 = 64$ cálculos. Mas se for explorada a estrutura local, pode-se diminuir o somatório para $2^3 + 2^3 = 16$ cálculos.

$$P(D) = \left(\sum_{A,B,C} P(A).P(B).P(C|A).P(D|A,B) \right) \left(\sum_{E,F,G} P(E).P(F|D).P(G|D,E) \right)$$

- Para um MBDR, sejam R e S dois esquemas, eles podem formar uma esquema maior através das operações de: produto cartesiano $R \times S$; junção theta $R \bowtie_{\theta} S$ (operação que combina produto cartesiano \times e seleção σ numa só operação, ou seja, é equivalente à $\sigma_{\theta}(R \times S)$ [KOR 89,p.65]); e, junção natural $R \bowtie S$ (é uma junção theta que força a igualdade naqueles atributos que aparecem em ambos os esquemas da relação [Ibid,p.66]).

Durante esta reconstrução, operações básicas de otimização incluem, por exemplo [Ibid,p.326-331]: i) fazer operações de seleção σ antes de produto cartesiano \times ; ii) transformar operações do tipo $\sigma_{p \text{ and } q}(\text{expressão_relacional})$ em $\sigma_p(\sigma_q(\text{expressão_relacional}))$; iii) seja $R \bowtie S \dots \bowtie Z$ uma seqüência de junções naturais, então explorar as propriedades associativas e comutativas destas junções para produzir uma ordenação ótima, de tal forma que o tamanho dos resultados temporários sejam reduzidos; e, iv) fazer operações de projeção Π antes de produtos cartesianos \times e de junções \bowtie , para reduzir o tamanho das tabelas (deixando apenas os atributos que aparecem no resultado da consulta, ou os que são necessários para o processamento de operações subseqüentes).

Agora serão utilizados exemplos para ilustrar as analogias feitas entre Redes Probabilísticas e Modelos de Banco de Dados Relacional. A Figura 2.17-A mostra uma Rede Probabilística que corresponde à uma Conexão Serial, e a Figura 2.17-B mostra uma representação alternativa desta rede num modelo conceitual de banco de dados relacional, juntamente com as tabelas. Note que, os valores 1 e 0 da coluna $B.B$ (da tabela da relação entre B e C), na verdade, são ponteiros para as respectivas colunas da tabela da Entidade B : $P(B|A)$ e $P(\bar{B}|A)$ – esta última coluna não está representada na tabela da Entidade B , mas é obtida calculando $1 - P(B|A)$.

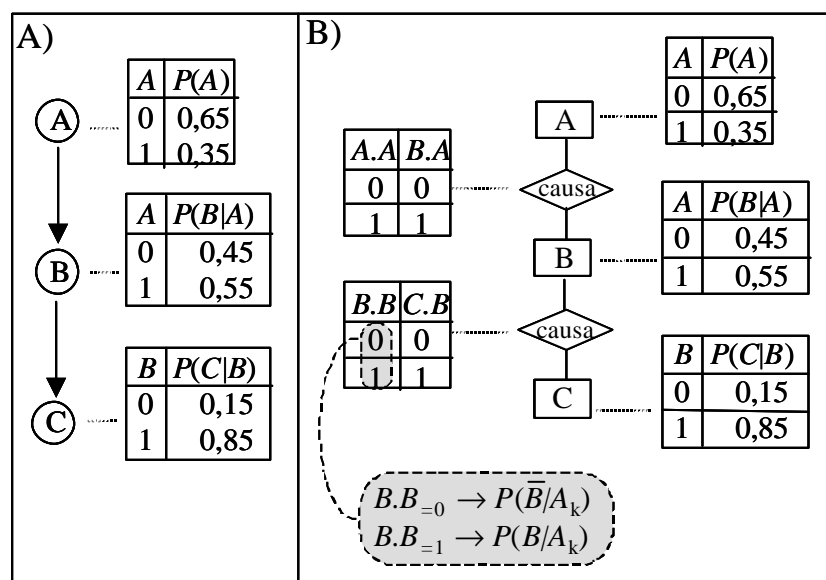


Figura 2.17: A) Exemplo de Rede Probabilística. B) Modelo de Entidades e Relacionamentos e Respetivas Tabelas

Inicialmente, será mostrada a Inferência Causal para descobrir $P(C_{=1}|A_{=1})$.

$$P(C_{=1} | A_{=1}) = \sum_B P(C_{=1}, B | A_{=1}) = \sum_B P(C_{=1} | B).P(B | A_{=1})$$

Então, substituindo os valores, e desenvolvendo o somatório

$$\begin{aligned}
 P(C_{=1}|A_{=1}) &= P(C_{=1}|B_{=0}) \cdot P(B_{=0}|A_{=1})^* + \\
 &\quad P(C_{=1}|B_{=1}) \cdot P(B_{=1}|A_{=1}) \\
 &= 0,15 \cdot (1 - 0,55) + \\
 &\quad 0,85 \cdot 0,55 \\
 &= 0,535
 \end{aligned}$$

*Observação:

$$P(B_{=0}|A_{=1}) = 1 - P(B_{=1}|A_{=1})$$

A Figura 2.18-A mostra o diagrama desta inferência e a Figura 2.18-B ilustra como as tabelas das entidades C, B e A (da Figura 2.17-B) são relacionadas.

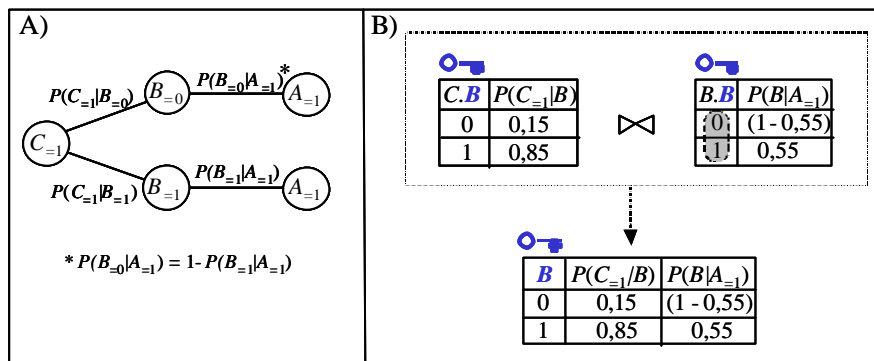


Figura 2.18: Exemplo 1: A) Diagrama da Inferência. B) Relacionamento das Tabelas

Ainda usando o exemplo da Figura 2.17, agora será mostrada a Inferência de Diagnóstico para descobrir $P(A_{=1}|C_{=1})$. Pela Regra de Bayes, temos

$$P(A_{=1} | C_{=1}) = \frac{P(C_{=1} | A_{=1}) \cdot P(A_{=1})}{P(C_{=1})}$$

Resta calcular $P(C_{=1})$, pois $P(A_{=1})$ está disponível e $P(C_{=1}|A_{=1})$ foi calculado anteriormente.

$$\begin{aligned}
 P(C_{=1}) &= \sum_B P(C_{=1} | B) \cdot \left[\sum_A P(B | A) \cdot P(A) \right] \\
 &= P(C_{=1}|B_{=0}) \cdot P(B_{=0}|A_{=0}) \cdot P(A_{=0}) + \\
 &\quad P(C_{=1}|B_{=0}) \cdot P(B_{=0}|A_{=1}) \cdot P(A_{=1}) + \\
 &\quad P(C_{=1}|B_{=1}) \cdot P(B_{=1}|A_{=0}) \cdot P(A_{=0}) + \\
 &\quad P(C_{=1}|B_{=1}) \cdot P(B_{=1}|A_{=1}) \cdot P(A_{=1}) \\
 &= 0,4895
 \end{aligned}$$

Portanto, $P(A_{=1}|C_{=1})=0,3825$. A Figura 2.19-A mostra o diagrama desta inferência e a Figura 2.19-B ilustra como as tabelas das entidades C, B e A (da Figura 2.17-B) são relacionadas.

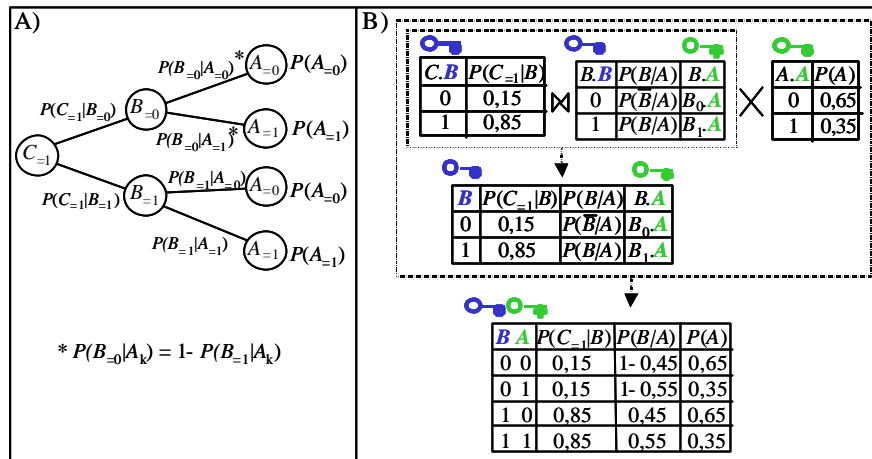


Figura 2.19: Exemplo 2: A) Diagrama da Inferência. B) Relacionamento das Tabelas

2.7.1 Dificuldade do Modelo Relacional para Controlar Semântica

Uma vez feita a analogia entre Rede Probabilística (RP) e o Modelo de Banco de Dados Relacional (MBDR), vamos agora mostrar uma comparação entre o MBDR com o Modelo de Banco de Dados Orientado a Objetos (MBDOO). Nesta comparação, subsidiada por [STO 96, LOO 95], vamos expor uma dificuldade do MBDR com relação ao controle da semântica dos dados.

Com relação aos domínios complexos, o uso de Abstração de Dados no MBDOO simplifica a modelagem, fazendo que um esquema do MBDOO seja bem mais simples que um esquema do MBDR. Além disso, os relacionamentos de um MBDOO são implementados usando identificadores de objetos (*ObjectIDs*) numa estrutura de grafo que propicia uma representação direta para os programas de aplicação. Já no MBDR, as instâncias dos esquemas relacionais devem ter um atributo identificador (chave) e os relacionamentos são implementados usando estas chaves em tabelas de relação.

A Herança é outra característica intrínseca ao MBDOO, permitindo definir uma classe em termos de outra. O MBDR não possui este recurso, sendo necessário acrescentar uma coluna de tipos na relação principal e então diferentes relações são criadas para cada tipo, uma vez que elas recebem informações diferentes. Através da Herança no MBDOO, fica mais simples percorrer a estrutura dos dados e manter o controle da semântica das instâncias dos objetos, via os *ObjectsIDs* – que são naturais para as linguagens de programação orientadas a objetos. Todavia, para os programas de aplicação do MBDR, deve-se fazer esforços extras para reunir os dados armazenados nas relações e também para entender a semântica dos tipos das tuplas – para que possa ser determinado o código apropriado a ser executado. Estes esforços extras são também conhecidos como descasamento de impedância (*impedance mismatch*) [PHI 91]. O MBDOO, por sua vez, entende a semântica dos tipos através do uso de classes e assim sabe qual é o código apropriado a ser executado. Além disso, um mesmo método pode ser usado em mais de uma classe, fazendo a operação de sobrecarga do método (*method overload*), sendo que o encapsulamento dos objetos garante que os dados sejam acessados somente pelos métodos corretos.

2.7.2 Dificuldade da Rede Probabilística para Controlar Semântica

Após concluir a comparação entre MBDR e MBDOO, vamos agora mostrar que as RP também apresentam uma dificuldade com relação ao controle da semântica, principalmente durante os procedimentos de inferência probabilística.

Os principais métodos de Inferência Probabilística (desde os clássicos apresentados em [PEA 88] e suas derivações posteriores apresentadas em conferências anuais, por exemplo *Conference on Uncertainty in Artificial Intelligence*¹), a grande maioria destes métodos usa uma operação básica chamada Marginalização (e pode ser facilmente percebida através dos somatórios Σ presentes nos algoritmos). Durante a Marginalização, existe uma acumulação de valores nos somatórios, todavia esta acumulação representa também uma sobreposição/construção dos caminhos causais, conforme pode ser visualizado nos diagramas de árvore que representam as inferências na medida em que os fluxos dos cálculos se propagam das folhas em direção às raízes. A Figura 2.20 ilustra estes fluxos de acumulação (onde as setas, neste caso, representam o sentido da acumulação, e não necessariamente a orientação dos relacionamentos causais).

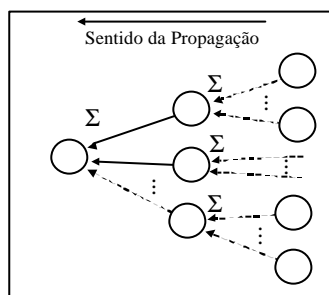


Figura 2.20: Operações de Marginalização

Devido ao fato da Rede Probabilística (RP) ser uma representação compacta do domínio modelado, assim como o Modelo de Bancos de Dados Relacional (MBDR), ela também apresenta uma dificuldade para controlar a semântica – assim como o MBDR, porque durante a Inferência Probabilística, a operação de Marginalização acumula somente valores numéricos oriundos dos caminhos envolvidos (valores estes que possuem uma semântica causal), mas não representam em si mesmos uma semântica de hierarquia de classes, da mesma forma que é representada no Modelo de Bancos de Dados Orientado a Objetos.

Apesar de novas abordagens combinarem orientação a objetos com Redes Probabilísticas, os fundamentos básicos de funcionamento das Redes Probabilísticas continuam os mesmos. Por exemplo, em [KOL 97], é descrita uma linguagem de Rede Probabilística (/Bayesiana) Orientada a Objetos (RPOO), mas ainda assim são usados fragmentos de Redes Probabilísticas para descrever as relações probabilísticas entre os atributos de um objeto. Por outro lado, [LAN 2001] propõe um método para aprendizagem de parâmetros para a RPOO – porque esta ainda preserva o problema de calcular e manter as estimativas de probabilidade, visto que algoritmos de aprendizagem como [6 Apud LAN 2001] não exploram o modelo orientado a objetos do domínio durante a aprendizagem. Mais recentemente, [LAN 2003] propõe um método para fazer aprendizagem estrutural em domínios orientados a objetos, estendendo o trabalho anterior [LAN 2001].

¹ www.auai.org

De um modo geral, o processo de aprendizagem das Redes Probabilísticas lida com um número exponencialmente grande de combinações resultantes do número de variáveis, do número de estados de cada variável, e do número de combinações dos nodos pais (os que causam os nodos filhos). Se for usado o método da força bruta, o processo de aprendizagem é da classe de complexidade NP-HARD [CHI 94]. Este custo computacional tem sido diminuído através de novas técnicas de aprendizagem tais como as apresentadas em [CAS 97, JEN 2001]. Posteriormente, técnicas de aprendizagem incremental para Redes Probabilísticas Dinâmicas foram apresentadas, por exemplo [GHA 98]. Mais recentemente, [SUT 2003] apresenta uma arquitetura para interpretar dados temporais e construir de forma incremental as Redes Probabilísticas, combinando a arquitetura de *Blackboard* com técnicas recentes de construção de Redes Probabilísticas a partir de fragmentos.

Observando estes exemplos representativos, pode-se perceber que, tanto a integração da orientação a objetos com as Redes Probabilísticas, quanto a introdução de aprendizagem incremental para as Redes Probabilísticas, ambas preservam o núcleo básico do modelo da Rede Probabilística. Este núcleo básico consiste numa representação compacta das relações causais do domínio, sendo que durante os processos de Inferência Probabilística, os caminhos dos relacionamentos causais (entre as variáveis envolvidas numa consulta) são reconstruídos durante o processo de propagação do impacto das evidências observadas. Todavia, durante esta reconstrução de caminhos, a operação básica de Marginalização acumula somente valores numéricos oriundos dos caminhos envolvidos (valores estes que possuem uma semântica causal), mas estes valores numéricos não representam em si mesmos uma semântica de hierarquia de classes.

2.8 Conclusões

Conforme exposto ao longo deste capítulo e analisado na seção anterior, as Redes Probabilísticas (ou Bayesianas) possuem uma dificuldade para controlar uma semântica de hierarquia de classes. Esta dificuldade cria o que pode ser definido como o **Problema da Lacuna** – o qual faz perder um grande número de relacionamentos semânticos valiosos (dos contextos de origem) enquanto os dados são armazenados para serem usados pelos processos de aprendizagem (da estrutura e dos parâmetros da Rede Probabilística). Esta perda de relacionamentos semânticos cria um tipo de dualismo, ou melhor, um *trade-off* entre as necessidades de: i) Coletar dados suficientes (visando obter importância estatística); *versus*, ii) Rastrear (ou acompanhar) problemas não-estacionários, o que é normalmente feito com técnicas de esmaecimento (por exemplo, *Fading*, em [JEN 2001,p.89]).

Portanto, a redução do Problema da Lacuna necessita de um sistema que aprenda durante a execução – o que depende do apoio de uma infraestrutura eficiente (escalável e com operações contínuas e incrementais) que possua uma forte organização contextual. Esta organização precisa possibilitar o controle da semântica de hierarquias de classes e também da semântica das relações de causa-e-efeito entre: percepções dos objetos do ambiente, percepções das ações (sendo realizadas sobre, ou relacionadas com os objetos do ambiente) e percepções da realimentação (*feedback*) dos efeitos ou resultados destas ações.

Além disso, se for considerado um ambiente incerto, dinâmico e complexo, existe uma necessidade de integrar os processos de percepção do ambiente, tomada de decisão e aprendizagem, porque somente um sistema completamente integrado proporciona a sinergia necessária para lidar com os problemas relacionados.

3 SERVIÇOS PARA AUXILIAR DECISÃO MEDIANTE INCERTEZA

O capítulo anterior mostrou que as Redes Probabilísticas fazem uma representação completa e compactada de uma Distribuição de Probabilidades Conjunta (DPC). Esta compactação decorre da identificação e codificação de uma coleção de declarações de independências condicionais entre as Variáveis Aleatórias do domínio modelado, criando um sistema localmente estruturado. Todavia, foi mostrado também que as Redes Probabilísticas possuem uma dificuldade para controlar uma semântica de hierarquia de classes, gerando o Problema da Lacuna (Seção 2.8).

Este capítulo descreve uma representação de conhecimento alternativa às Redes Probabilísticas, ao propor Serviços para Auxiliar Decisão mediante Incerteza (SADI). Esta nova representação possui a semântica causal e também a semântica de hierarquia de classes (da orientação a objetos). Assim como ocorre no ser humano (mais precisamente, no Sistema Psi ψ [MAC 2003,p.31]), o SADI não representa todas as variações teoricamente possíveis dos estados das Variáveis Aleatórias (isto é, o SADI não faz uma representação completa de uma DPC, mas sim uma representação aproximada de uma DPC). O SADI se baseia na delimitação dos contextos tanto do ponto de vista físico quanto comportamental para criar um sistema localmente estruturado. A identificação de contextos é uma tarefa de classificação/filtragem, para isto o SADI se baseia na arquitetura de serviços SAIC [BOR 98], estendendo-a e adaptando-a para operar com as seguintes condições: é possível perceber o ambiente, fazer decisões mediante incerteza, perceber a realimentação (*feedback*) referente ao sucesso ou fracasso das ações, e aprender com a experiência.

Este capítulo está organizado da seguinte forma: a Seção 3.1 define o problema a ser resolvido, assim como o objetivo e o ambiente deste trabalho; a Seção 3.2 descreve a solução para o problema definido; na Subseção 3.2.1, o *Framework* SAIC é revisado (porque ele serve de base para a solução apresentada); nas Subseções 3.2.2, 3.2.3 e 3.2.4 o *Framework* SADI é então descrito, juntamente com a prova teórica de que as propriedades do *Framework* SAIC são preservadas.

3.1 Descrição do Problema

Definição 1.1 O Problema

Dado um *Framework* de Comunicação (denominado SAIC, o qual significa *Services to Aid Information Consumption* [BOR 97,BOR 98]), deseja-se construir um novo *Framework* de Decisão (denominado SADI, o qual significa *Serviços para Auxiliar Decisão mediante Incerteza*), onde os serviços SADI usam os serviços SAIC, e o *Framework* SADI deve preservar as propriedades do *Framework* SAIC: serviços escaláveis, incrementais e contínuos.

Mais especificamente, os serviços SAIC de Busca, Filtragem e Disseminação são usados para construir o *Framework SADI*, que é composto de:

- Uma Arquitetura de Serviços que integra:
 - Percepção do Ambiente;
 - Detecção de Contexto;
 - Aprendizagem;
 - Tomada de Decisão;
- Uma Representação de Conhecimento que integra:
 - Relacionamentos Causais;
 - Relacionamentos Semânticos Multidimensionais;
 - Estrutura de Acesso aos Dados.

Esta Representação de Conhecimento é construída através da aprendizagem de relações de causa-e-efeito entre a percepção de objetos do ambiente e da percepção de ações sendo realizadas sobre (ou relacionadas com) os objetos do ambiente.

Definição 1.2 O Objetivo

O *Framework SADI* objetiva usar a Arquitetura de Serviços para construir e manter a Representação de Conhecimento para aplicações que necessitam de:

- Inferência em Tempo-Real (diagnóstico e prognóstico);
- Aprendizagem em Tempo-Real (adaptação do modelo às mudanças do sistema).

O *Framework SADI* assume que as aplicações descrevem os objetos do ambiente, as ações a serem realizadas nestes objetos, assim como as recompensas (*reward functions*, que medem o sucesso ou insucesso de ações sobre objetos).

Definição 1.3 O Ambiente

O *Framework SADI* é projetado para operar em sistemas complexos, dinâmicos e distribuídos. Aqui, o **Ambiente**² destes sistemas distribuídos pode ser: Um *Ambiente Físico*; ou, um *Ambiente Virtual*. Ainda é considerado também que:

- Um sistema distribuído num *Ambiente Virtual* pode ser: um modelo simulado (de um sistema distribuído num *Ambiente Físico*);
- Um *Ambiente Virtual* pode exibir: comportamento e características similares aos sistemas distribuídos num *Ambiente Físico*.

3.2 Solução do Problema

Primeiro, na Subseção 3.2.1, é revisado o *Framework SAIC* (e suas propriedades principais). Depois, nas Subseções 3.2.2, 3.2.3 e 3.2.4, o *Framework SADI* é descrito (isto é, a *Arquitetura de Serviços* e a *Representação de Conhecimento*), juntamente com a prova teórica de que o *Framework SADI* preserva as propriedades do *Framework SAIC*, que são: serviços escaláveis, incrementais e contínuos.

3.2.1 O *Framework SAIC*

² Ao longo deste capítulo, a definição de um conceito está realçada em **negrito**, sendo que qualquer referência posterior a um conceito já definido está realçada em *itálico*.

Definição 2.1 A Arquitetura de Serviços SAIC

Os Serviços para Auxiliar Consumo de Informação (SAIC) [BOR 97, BOR 98] são:

- Busca (Definição 2.1.1);
- Filtragem (Definição 2.1.2);
- Disseminação (Definição 2.1.3).

Os serviços de Busca, Filtragem e Disseminação criam Nichos de Informação similares ao espaço de informação dos Grupos de Notícias (*Newsgroups*), onde sistemas de aplicação podem desenvolver serviços usando as meta-informações de baixo custo disponíveis nos Nichos de Informação.

Definição 2.1.1 O Serviço de Busca

O **Serviço de Busca** detecta de forma incremental as mudanças (criação, remoção, modificação) de Provedores de Informação (dentro de um ambiente de rede) e produz Meta-Informação sobre estas mudanças. A Meta-Informação é composta pelo conteúdo mais representativo da informação; e, pelos dados resultantes da geração da Meta-Informação.

O Serviço de Busca é dividido por subdomínio de rede, onde um *Broker* armazena a Meta-Informação gerada no subdomínio. Para aumentar a eficiência, os Provedores de Informação executam o *Serviço de Busca* localmente.

O Nicho raiz da Hierarquia de Tópicos é distribuído e armazenado pelo conjunto de todos os *Brokers* de subdomínio de rede.

Um novo Nicho de Informação pode ser uma especialização de outro Nicho já existente e mais genérico, ou é um Nicho de primeiro-nível na Hierarquia de Tópicos. Cada novo Nicho possui seu próprio *Broker* (ver Figura 3.1).

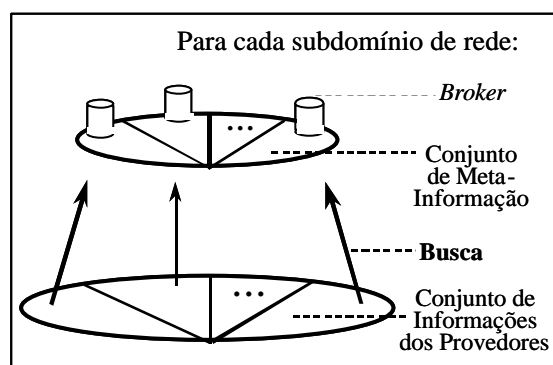


Figura 3.1: O Serviço de Busca

Definição 2.1.2 O Serviço de Filtragem

O **Serviço de Filtragem** recebe Meta-Informação do *Serviço de Busca* e faz roteamento por contudo, isto é, filtra de forma incremental estas Meta-Informação para seus Nichos de Informação apropriados (onde cada Nicho usa um *Broker* para armazenagem). Os Nichos de Informação são organizados logicamente numa Hierarquia de Tópicos.

Quando um novo Nicho é criado, o seu *Broker* registra uma Consulta de Filtragem no *Broker* do Nicho mais genérico. Se este novo Nicho é um de primeiro-

nível na Hierarquia de Tópicos, então o seu *Broker* registra uma Consulta de Filtragem em cada *Broker* de subdomínio de rede (ver Figura 3.2)

O Teorema 2.1.1 (abaixo) prova que o *Serviço de Filtragem* é mais eficiente que o *Serviço de Busca*, preservando assim os custos incrementais do *Serviço de Busca*.

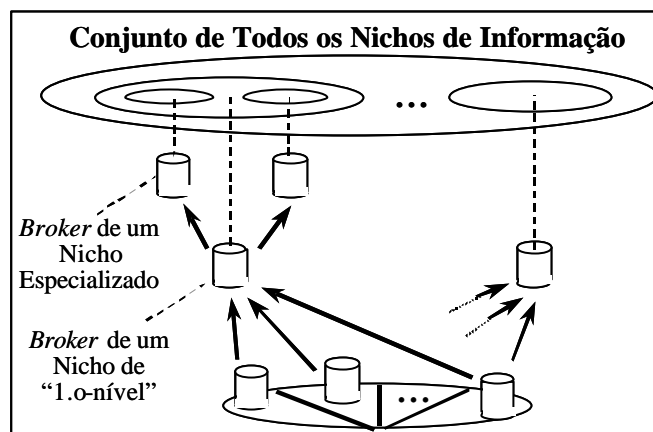


Figura 3.2: O Serviço de Filtragem

Teorema 2.1.1 O registro de Consultas de Filtragem (em *Brokers* de Nichos mais genéricos) é mais eficiente (em termos de custos de: seleção, re-indexação e frequência de re-indexação) que a submissão de Consultas de Pesquisa (de um *Broker* de Nicho mais específico, para um *Broker* de Nicho mais genérico) [BOR 97, BOR 98].

PROVA:

Pesquisa e Filtragem são um tanto simétricas: no caso da Pesquisa, uma consulta é recebida e conferida com um índice de documentos (ou índice de objetos de informação, p.ex. Meta-Informação); no caso da Filtragem, um documento é recebido e conferido com um índice de consultas.

Duas características importantes do cenário dos Nichos de Informação determinam a eficiência da operação de Filtragem: 1.o) A hierarquia de Nichos é quase estável (assim como a hierarquia dos *Newsgroups*); e, 2.o) O Serviço de Busca é incremental.

Quando uma Hierarquia de Tópicos específicos é construída com consultas de Pesquisa (p.ex. O Sistema *Harvest* [Bowman *et al*, apud BOR 98]), os *Brokers* de tópicos mais específicos submetem consultas de Pesquisa aos *Brokers* de tópicos mais genéricos. Considerando a situação ilustrada na Figura 3.3-a, o custo total do subsistema *Index/Search* do Sistema *Harvest* é composto de dois custos principais: 1.o) O custo de selecionar, no conjunto indexado de $(M+N)$ Meta-Informação no *Broker* de tópico mais genérico, aquelas que casam com cada uma das K consultas submetidas; 2.o) O custo de re-indexar o conjunto de M Meta-Informação armazenado no *Broker* de tópico mais genérico, para incorporar as N Meta-Informação recém-chegadas.

Na construção da hierarquia de Nichos SAIC, usando consultas de Filtragem, os *Brokers* de Nichos mais especializados (*BrokerN+S*) registram suas consultas de Filtragem no *Broker* de Nicho mais genérico (*BrokerN+G*). Considerando a situação ilustrada na Figura 3.3-b, o custo total do *Serviço de Filtragem* SAIC é composto por dois custos principais: 1.o) O custo de selecionar, no conjunto indexado de $(K+I)$ consultas de Filtragem armazenadas no *BrokerN+G*, aquelas que casam com cada uma das N Meta-Informação recém-chegadas; 2.o) O custo de re-indexar o conjunto de K

consultas armazenadas no BrokerN+G, para incorporar as I consultas de Filtragem recém-chegadas (espera-se que I seja muito pequeno, portanto é considerado uma constante).

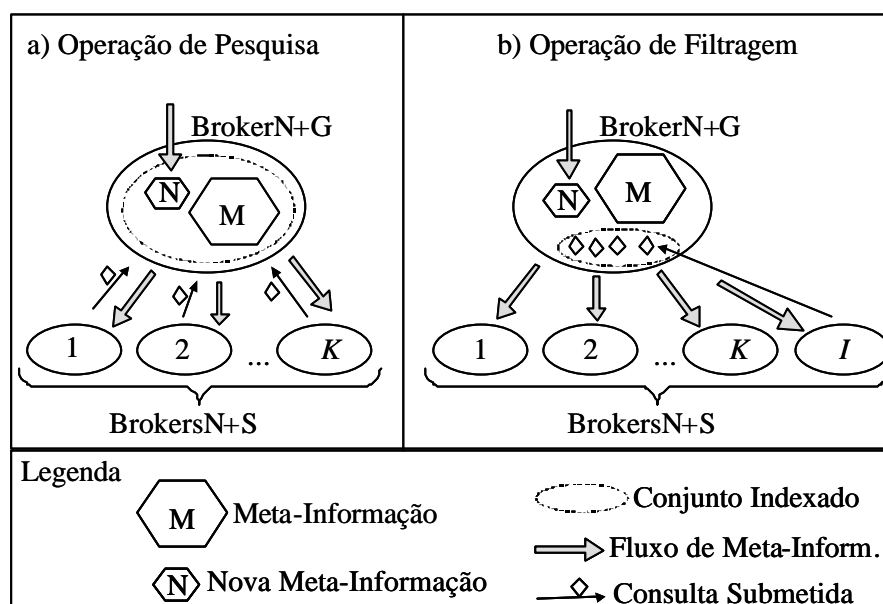


Figura 3.3: a) Operação de Pesquisa; b) Operação de Filtragem

As hierarquias (onde ocorrem as operações de Pesquisa e Filtragem) recebem Meta-Informação de forma incremental. Em ambas hierarquias, a célula básica é uma subárvore (como mostrado nas Figuras 3.3-a e 3.3-b). A operação de Filtragem é mais eficiente que a operação de Pesquisa porque:

- O custo total da operação de Filtragem $O(K \times N)$ é menor que o custo total da operação de Pesquisa $O(K \times (M+N))$;
- O custo de re-indexação para atualizar a base de consultas de Filtragem $O(K)$ é menor que o custo de re-indexação para atualizar a base de Meta-Informação $O(M+N)$;
- A frequência de re-indexação da base de consultas de Filtragem é menor que a frequência de re-indexação da base de Meta-Informação, porque é esperado que a criação de um novo Nicho seja esporádica, enquanto que a chegada de novas Meta-Informação num Nicho é esperado que seja quase contínua.

Definição 2.1.3 O Serviço de Disseminação

O **Serviço de Disseminação** consiste da replicação, *caching* e transmissão de fluxos de informação. O *Serviço de Disseminação* preserva os custos incrementais dos *Serviços de Busca e Filtragem*, mesmo com a existência de réplicas de *Brokers* (isto é, a Hierarquia de Tópicos pode estar fisicamente distribuída).

Cada *Broker* de Nicho de Informação pode ter um grupo de réplicas distribuídas (uma réplica por subdomínio de rede).

A Transmissão “Vertical” é a transmissão de Meta-Informação, filtrada num Nicho mais genérico, para seus Nichos mais específicos. Esta transmissão ocorre entre *Brokers* que estão fisicamente próximos (ver Figura 3.4-a).

A Transmissão “Horizontal” é a transmissão de Meta-Informação, que um *Broker* de Nicho mais específico recebeu pela Transmissão “Vertical”, para suas réplicas no mesmo Nicho de Informação (ver Figura 3.4-b).

O Lema 2.1.1 (abaixo) demonstra como as Transmissões “Vertical” e “Horizontal” preservam os custos incrementais dos *Serviços* de *Busca* e de *Filtragem*, mesmo quando a hierarquia de Nichos contém *Brokers* com réplicas distribuídas.

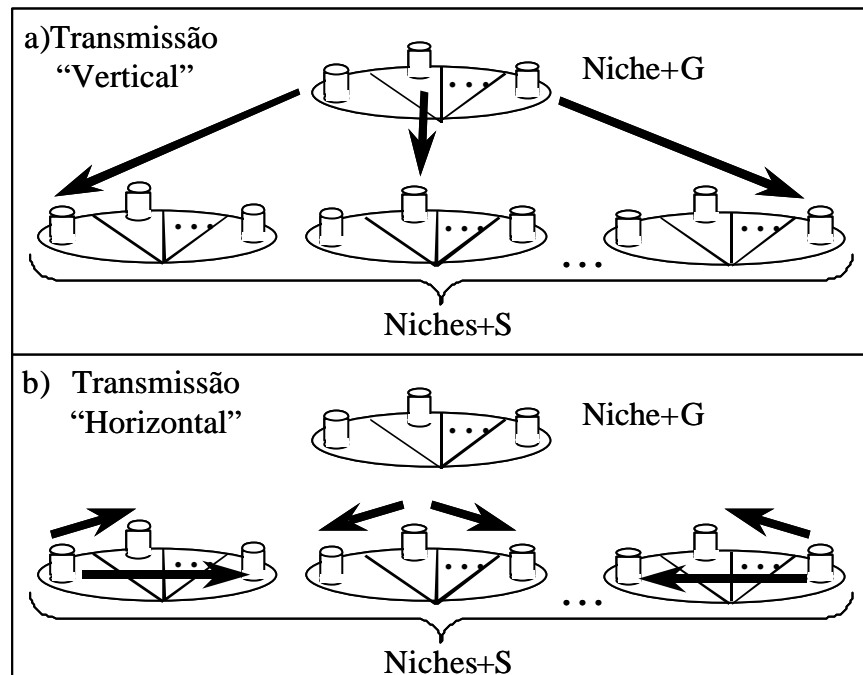


Figura 3.4: Transmissões “Vertical” e “Horizontal”

Lema 2.1.1 A Transmissão “Vertical” (TV) e a Transmissão “Horizontal” (TH) otimizam o *Serviço de Filtragem* e o fluxo de Meta-Informação na rede.

PROVA:

Numa sub-árvore da Hierarquia de Nichos, a TH (Figura 3.4-b) em cada Nicho+S minimiza o *Serviço de Filtragem* SAIC no Nicho+G, porque o Nicho+G necessita armazenar apenas uma consulta de Filtragem para cada Nicho+S. Portanto, o Nicho+G realiza apenas uma TV (Figura 3.4-a) por Nicho+S, não importando quantas réplicas de *Brokers* existem em cada Nicho+S.

Adicionalmente, a TH no Nicho+G distribui a carga do *Serviço de Filtragem* SAIC entre as réplicas de *BrokerN+G* que pertencem a este Nicho+G, porque cada réplica de *BrokerN+G* armazena a consulta de Filtragem do *BrokerN+S* que está fisicamente mais próximo (como ilustrado na Figura 3.4-a). Portanto, a TV ocorre entre réplicas de *BrokerN+G* e *BrokerN+S* que estão fisicamente mais próximos (no caso ideal, eles pertencem ao mesmo subdomínio de rede).

3.2.2 Framework SADI – Arquitetura de Serviços

Definição 2.2 A Arquitetura de Serviços SADI

Os **Serviços para Auxiliar Decisão mediante Incerteza** (SADI) são:

- Percepção do Ambiente (Definição 2.2.1);
- Detecção de Contexto (Definição 2.2.2);
- Tomada de Decisão (Definição 2.2.3); e,
- Aprendizagem (Definição 2.2.4).

A Arquitetura de Serviços SADI constrói e mantém de forma incremental a Representação de Conhecimento (Definição 2.3), através da execução do seguinte ciclo de operação:

- 1^o) Percepção do Ambiente (Percepções são introduzidas);
- 2^o) Detecção de Contexto (Um ou mais Contextos são ativados; e, percepções são reconhecidas);
- 3^o) Aprendizagem (O *Feedback* é processado, dado as percepções reconhecidas nos Contextos ativados);
- 4^o) Tomada de Decisão (Ações são decididas e realizadas, dado os Contextos ativados).

Observação 1) Este é um ciclo de operação contínuo, porque as percepções induzem ações e as ações produzem novas percepções. Este ciclo de operação é implementado como um *pipeline*, e os 3^o e 4^o passos podem ser executados em paralelo se eles ocorrem em Contextos (Definição 2.2.2.1) diferentes. Este processamento paralelo pode ser realizado mesmo se os diferentes Contextos possuem Símbolos (Definição 2.2.1.1) em comum, porque cada Contexto possui suas próprias réplicas de Símbolos, sua própria semântica, e sua própria contagem lógica de tempo.

Observação 2) Conforme mostrado no ciclo acima, o Serviço de Aprendizagem (Definição 2.2.4) ocorre antes do Serviço de Tomada de Decisão (Definição 2.2.3). A razão para esta seqüência se torna clara durante a definição destes serviços.

Definição 2.2.1 Serviço de Percepção do Ambiente

O **Serviço de Percepção do Ambiente (SPA)** é similar ao *Serviço de Busca* (SB) SAIC (Definição 2.1.1). Todavia, o SPA aplica a funcionalidade do SB ao conceito mais amplo de *Ambiente* (Definição 1.3).

Com relação ao SPA, este conceito de *Ambiente* é especializado aqui: *Ambiente* é também o lugar onde percepções são originadas e o lugar onde ações são realizadas. Dentro deste *Ambiente*, uma entidade que percebe e age é definida aqui como um **Corpo**, e este *Corpo* pode ser uma entidade física (p.ex. ser vivo ou máquina) ou uma entidade puramente virtual (p.ex. agente de software). Dentro deste *Ambiente*, as percepções podem ser:

- **Estímulo Interno:** aquelas percebidas no interior do *Corpo*; ou
- **Estímulo Externo:** aquelas percebidas no exterior do *Corpo*.

Além disso, o SPA estende o conceito de Meta-Informação (produzida pelo SB). Este conceito estendido é definido aqui como Símbolo, a seguir.

Definição 2.2.1.1 Símbolo

É assumido que percepções são recebidas como padrões gerados por sensores específicos que capturam estímulos (como imagens, sons, temperatura, etc), ou percepções são recebidas como *tags/tokens* (incluindo Meta-Informação) extraídas por *parsers* específicos, etc.

É assumido também que estes sensores capturam estímulos, formam padrões e os classifica em hierarquias. Este padrão classificado/reconhecido é definido aqui como **Símbolo**.

Para cada hierarquia de classificação, cada sub-árvore inicia com um *Símbolo* mais genérico/abstrato na raiz, e especializa os *Símbolos* na medida que os nodos são expandidos.

Definição 2.2.1.2 Símbolo Composto

Símbolo Composto é a combinação de dois ou mais *Símbolos* (Definição 2.2.1.1) que são percebidos simultaneamente.

Definição 2.2.1.3 Tempo de Sistema T_S

Tempo de Sistema T_S é a contagem lógica de tempo usada para sincronizar a propagação de percepções: *Símbolos* (Definição 2.2.1.1) e *Símbolos Compostos* (Definição 2.2.1.2). Um T_S corresponde a um ciclo de operação do *Framework* SADI (isto é, o ciclo: Percepção do Ambiente, Detecção de Contexto, Aprendizagem e Tomada de Decisão).

Definição 2.2.1.4 Entrada de Percepção

A **Entrada de Percepção** é o conjunto de *Símbolos* recebidos num dado *Tempo de Sistema T_S* e possui a propriedade do **Uso Duplo**:

- Alimentar adiante (*Feedforward*) a ativação de um ou mais Contextos (Definição 2.2.2.1) realizada em T_S pelo Serviço de Detecção de Contexto (Definição 2.2.2); e,
- Realimentar (*Feedback*), para cada Contexto (Definição 2.2.2.1) ativado em T_S , os possíveis efeitos/resultados das ações realizadas em algum ciclo de operação anterior do SADI, isto é ações realizadas em T_{S-t} (onde $t = 1, 2, \dots, n$). Este *Feedback* é usado pelo Serviço de Aprendizagem (Definição 2.2.4).

Definição 2.2.1.5 Espaço de Percepção

As hierarquias de percepção de todos os sensores compõe o **Espaço de Percepção** multidimensional. A *Entrada de Percepção* (Definição 2.2.1.4) é propagada dentro do *Espaço de Percepção*.

Para dar suporte às operações incrementais do *Framework* SADI, o *Espaço de Percepção* é subdividido em subespaços, e três destes subespaços possuem um relacionamento especial:

1. Um subespaço para as **Percepções de Objetos do Ambiente (PO)**;
2. Um subespaço para as **Percepções de Ações (PA)** sendo realizadas sobre (ou relacionadas com) os *PO*; e

3. Um subespaço para as **Percepções de Feedback Interior (PI)** – que são uma subclasse especial dos Objetos do *Ambiente*. O *PI* é um *Estímulo Interno* (em Definição 2.2.1) que mede a recompensa/reward de *PA* relacionada com *PO*.

Observação) Conforme descrito anteriormente no Objetivo do *Framework SADI* (Definição 1.2), o *Framework SADI* assume que as aplicações descrevem os Objetos do *Ambiente*, as ações a serem realizadas nestes Objetos, assim como as recompensas.

Definição 2.2.1.6 Fluxo de Percepção

Uma vez que a Representação de Conhecimento SADI (Definição 2.3) também possui Relacionamentos Semânticos Multidimensionais (Definição 2.3.2), e também que um *Símbolo* (Definição 2.2.1.1) percebido pode ocorrer em mais de um Contexto (Definição 2.2.2.1), então um *Símbolo* percebido é propagado/distribuído para cada dimensão de seus relacionamentos semânticos, em cada Contexto onde ocorrem.

Durante a propagação da *Entrada de Percepção* (2.2.1.4) dentro do *Espaço de Percepção* (2.2.1.5), um **Fluxo de Percepção** é o processo inteiro onde estímulos são capturados, padrões são formados e classificados em *Símbolos*, *Símbolos* são distribuídos para Contextos (Definição 2.2.2.1), e *Símbolos* menores percebidos são combinados em *Símbolos Compostos* (Definição 2.2.1.2) e então estes *Símbolos Compostos* são também distribuídos para Contextos.

Exemplo 2.2.1 Alguns *Símbolos* são mostrados numa sub-árvore da hierarquia de percepção de um sensor visual (Figura 3.5-A), e a Figura 3.5-B ilustra algumas hierarquias de percepção compondo o *Espaço de Percepção* multidimensional.

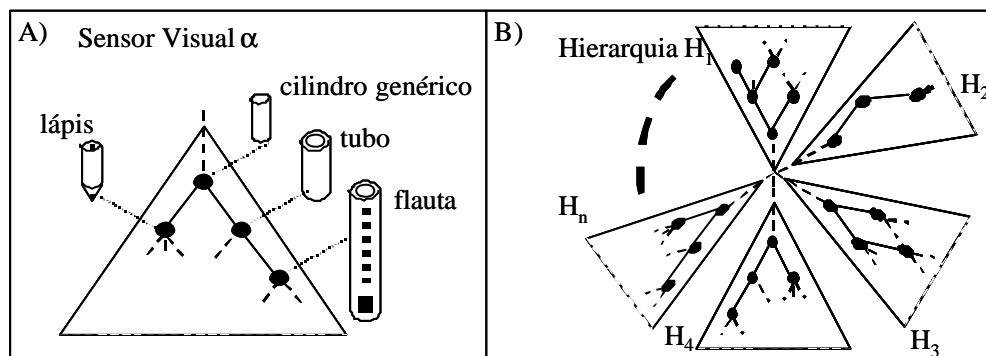


Figura 3.5: A) *Símbolos* numa sub-árvore de percepção; B) *Espaço de Percepção*

Definição 2.2.1.7 Percepção Certa

Um *Símbolo* (Definição 2.2.1.1) é uma **Percepção Certa** se ele já fornece sua sintaxe e semântica. Uma *Percepção Certa* possui um Contexto (Definição 2.2.2.1) bem definido e o Serviço de Detecção de Contexto (Definição 2.2.2) não precisa realizar o Casamento de Sintaxe e Semântica (Definição 2.2.2.5). Um *Símbolo* de um *Estímulo Interno* (em Definição 2.2.1) é sempre uma *Percepção Certa*.

Definição 2.2.1.8 Percepção Incerta

Um *Símbolo* (Definição 2.2.1.1) é uma **Percepção Incerta** quando ele apenas provê sua sintaxe. Portanto, uma *Percepção Incerta* necessita que Serviço de Detecção

de Contexto (Definição 2.2.2) realize o Casamento de Sintaxe e Semântica (Definição 2.2.2.5). Um *Símbolo* de um *Estímulo Externo* (em Definição 2.2.1) é sempre uma *Percepção Incerta*.

Exemplo 2.2.2 Sobre *Percepções Certas e Estímulo Interno*.

Alguém que sente fome (*Estímulo Interno*) não tem dúvida com relação a isto, porque a percepção (isto é, a sintaxe) e o contexto (isto é, a semântica) sobre a fome são automaticamente providos e associados pelo ambiente interno do corpo. Outro exemplo de *Estímulo Interno* pode ser um pedido de interrupção de um dispositivo de *hardware*.

Exemplo 2.2.3 Sobre *Percepções Incertas e Estímulo Externo*.

Um exemplo de *Estímulo Externo* pode ser uma imagem de um “terreno limpo” capturada por um sensor visual (p.ex. olho ou câmera). A *Percepção Incerta* “terreno” recebe dinamicamente sua semântica “limpo” do Serviço de Detecção de Contexto, a ser descrito agora.

Definição 2.2.2 Serviço de Detecção de Contexto

O **Serviço de Detecção de Contexto (SDC)** usa a funcionalidade do *Serviço de Filtragem (SF)* SAIC (Definição 2.1.2) e a do *Serviço de Disseminação (SD)* SAIC (Definição 2.1.3). O *SDC* também usa conhecimento do Modelo de Mundo (Definição 2.2.2.4).

O *SDC* aplica as funcionalidades do *SF* e *SD* para o conceito mais amplo de *Ambiente* (Definição 1.3, e estendido em Definição 2.2.1), o qual é também estendido aqui: o *Ambiente* também exhibe a propriedade descrita por [RUS 95, pp.441] como Sistema Esparço (também chamado Sistema Localmente Estruturado). Num **Sistema Esparso**, cada variável interage diretamente com apenas um número limitado de outras variáveis, não importa a quantidade total de variáveis (note que, aqui, variável é substituída pelo conceito de *Símbolo*, Definição 2.2.1.1).

Portanto, o *SDC* usa os *SF* e *SD* (com este conceito estendido de *Ambiente*) para alcançar dois objetivos simultâneos e complementares:

- Capturar os relacionamentos entre *Símbolos* percebidos e criar representações localmente estruturadas para estes relacionamentos capturados, através da organização destas representações em Contextos (Definição 2.2.2.1);
- Definir dinamicamente a semântica das *Percepções Incertas*, isto é, realizar o Casamento de Sintaxe e Semântica (Definição 2.2.2.5) – através do uso do conhecimento do Modelo de Mundo (Definição 2.2.2.4).

Definição 2.2.2.1 Contexto

O conceito de Contexto inclui (e estende) a definição de Redes Semânticas [RIC 93, pp.291] (basicamente, numa Rede Semântica, o significado de um conceito depende de como este conceito está conectado com outros conceitos). Portanto:

- Um **Contexto** é um conjunto de *Símbolos* (Definição 2.2.1.1) cujos relacionamentos definem a semântica, isto é, a semântica de um *Símbolo* depende de como este *Símbolo* está relacionado com outros *Símbolos*;
- Para cada *Entrada de Percepção* (Definição 2.2.1.4), um conjunto não-vazio de *Contextos* recebe o *Fluxo de Percepção* (Definição 2.2.1.6) através da operação combinada do *Serviço de Filtragem SAIC* (Definição 2.1.2) e o *Serviço de Disseminação SAIC* (Definição 2.1.3);
- Um *Contexto* tem a semântica de suas *Percepções Incertas* definida dinamicamente pelo Casamento de Sintaxe e Semântica (Definição 2.2.2.5) – esta operação de casamento usa o Modelo de Mundo (Definição 2.2.2.4);
- Um *Contexto* é ativado se e somente se: 1^a) ele recebe o *Fluxo de Percepção* (Definição 2.2.1.6); e, 2^a) suas *Percepções Incertas* são dinamicamente definidas;
- Um *Contexto* delimita o escopo dos relacionamentos e permite a descoberta de relacionamentos causais entre *Símbolos*, isto é, a descoberta de relações de causa-e-efeito entre *PO*, *PA* e *PI* (em Definição 2.2.1.5). Esta descoberta se baseia no novo conceito de Tempo Contextual (Definição 2.2.2.2).

Observação 1) Com relação à representação de relacionamentos causais (ou condicionais), a importância do contexto é também enfatizada por [PEA 88, pp.31], enquanto discorre sobre a declaração de probabilidade condicional $P(A|B)$ – que significa a probabilidade de *A*, dada a ocorrência de *B*. Também é declarado que: “os filósofos Bayesianos consideram o relacionamento condicional como sendo mais básico do que o de eventos conjuntos, isto é, mais compatível com a organização do conhecimento humano. Nesta visão, *B* serve como um ponteiro para um contexto ou *frame* de conhecimento, e $A|B$ significa o evento *A* no contexto especificado por *B*”.

Na Representação de Conhecimento do SADI (Definição 2.3), os Relacionamentos Causais também estão representados (Definição 2.3.1), e eles estão organizados em *Contextos*.

Observação 2) Enquanto discorre sobre a modelagem automática (a partir de uma perspectiva estatística), mais especificamente: a importância estatística *versus* a importância prática, [ELD 96] conclui que a introdução de contextos é útil para qualificar melhor as descobertas como sendo descobertas importantes. Outra conclusão apresentada é a dificuldade de incluir tal informação de contexto em sistemas de modelagem automática.

No *Framework SADI*, a estratégia para incluir informação de contextos (segundo a aceção de contexto de [ELD 96]) é:

- Definir dinamicamente a semântica de *Percepções Incertas* (Definição 2.2.1.8), isto é, realizar o Casamento de Sintaxe e Semântica (Definição 2.2.2.5) – esta operação de casamento usa o Modelo de Mundo (Definição 2.2.2.4);
- Realizar o Serviço de Aprendizagem (Definição 2.2.4) incremental enquanto realiza o Serviço de Tomada de Decisão (Definição 2.2.3) incremental. No Serviço de Aprendizagem é mostrada a importância do *Contexto* na redução da lacuna entre os processos de tomada de decisão e de aprendizagem. Esta redução da lacuna é também baseada no novo conceito de Tempo Contextual (abaixo).

Definição 2.2.2.2 Tempo Contextual T_C

O Tempo Contextual T_C especializa o conceito do *Tempo de Sistema* T_S (Definição 2.2.1.3). O **Tempo Contextual T_C** é a contagem lógica de tempo realizada por cada *Contexto* (Definição 2.2.2.1), sempre que um *Contexto* for ativado. O T_C estabelece a base conceitual para a operação incremental do Serviço de Aprendizagem (Definição 2.2.3).

O Lema 2.2.1 (abaixo) demonstra como o conceito de T_C expressa a relatividade da aprendizagem (dentro de *Contextos*).

Lema 2.2.1 O *Tempo Contextual T_C* (Definição 2.2.2.2) permite que a relação de causa-e-efeito seja relativa a cada *Contexto* (Definição 2.2.2.1) ativado.

PROVA:

Vamos rever e comentar a propriedade do *Uso Duplo* de uma *Entrada de Percepção* (Definição 2.2.1.4). A *Entrada de Percepção* (num dado *Tempo de Sistema* T_S , Definição 2.2.1.3) é usada simultaneamente para:

- Alimentar adiante (*Feedforward*) a ativação de um ou mais *Contextos*. Portanto, cada *Contexto* ativado incrementa seu T_C , e ações realizadas em T_C podem causar possíveis efeitos/resultados (a serem percebidos na próxima ativação de *Contexto*, isto é, em T_{C+1}).
- Realimentar (*Feedback*), para cada *Contexto* ativado (em T_C), os possíveis efeitos/resultados (das ações realizadas na ativação anterior do *Contexto*, isto é, em T_{C-1}). Note que, T_{C-1} não é necessariamente T_{S-1} , isto é, $T_{C-1} = T_{S-t}$ (onde $t = 1, 2, \dots, n$).

Dado que cada T_C é apenas incrementado quando seu respectivo *Contexto* é ativado, e uma vez que a *Entrada de Percepção* (num dado T_S) pode ativar um conjunto de *Contextos* (C_1, \dots, C_m) com seus respectivos contadores de *Tempo Contextual* (T_{C_1}, \dots, T_{C_m}), então a *Entrada de Percepção* (num dado T_S) pode trazer os possíveis efeitos/resultados de ações realizadas em ($T_{C_1-1}, \dots, T_{C_m-1}$) onde cada T_{C_i-1} ($i = 1, 2, \dots, m$) pode ser diferente de T_{S-1} .

Uma vez que a *Entrada de Percepção* usualmente varia, então o conjunto de *Contextos* ativados (C_1, \dots, C_m) também varia, portanto seus respectivos contadores de *Tempo Contextual* (T_{C_1}, \dots, T_{C_m}) podem ser diferentes um do outro (além da possibilidade de serem diferentes de T_{S-1}).

Exemplo 2.2.4 Sobre *Tempo Contextual T_C* .

Considerando a Figura 3.6, a *Entrada de Percepção* em $T_S = 9$ ativa os *Contextos* C_1 e C_n , e podem trazer os possíveis efeitos/resultados de ações realizadas no *Contexto* C_1 em $T_{C_1} = 2$ ($T_S = 3$), e ações realizadas no *Contexto* C_n em $T_{C_n} = 1$ ($T_S = 5$).

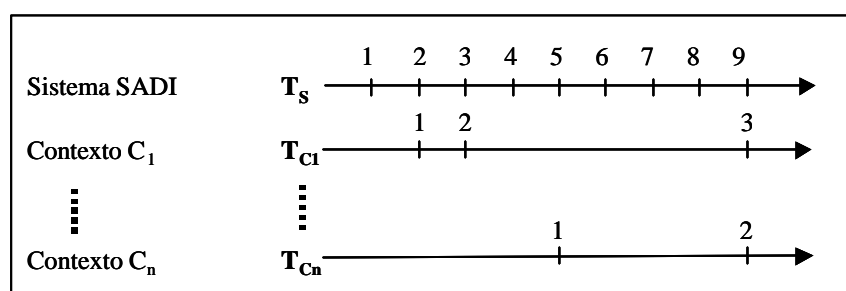


Figura 3.6: Exemplo de contagem de *Tempo Contextual*

Definição 2.2.2.3 Espaço de Contexto

O **Espaço de Contexto** é o conjunto de todos os *Contextos* (Definição 2.2.2.1). O *Espaço de Contexto* inclui o *Espaço de Percepção* (Definição 2.2.1.5) e o Modelo de Mundo (Definição 2.2.2.4).

O *Espaço de Contexto* contem o modelo conceitual que guia o crescimento da Representação de Conhecimento SADI (Definição 2.3), isto é, o *Espaço de Contexto* contem a união de: ^{1ª}) a sintaxe fornecida pelo *Espaço de Percepção*; e, ^{2ª}) a semântica fornecida pelo Modelo de Mundo.

O *Espaço de Contexto* é o ponto de entrada comum de onde todos os *Contextos Estendidos* (Definição 2.2.2.6) são derivados, porque cada *Contexto Estendido* é uma instância de um *Contexto* definido no *Espaço de Contexto*.

Adicionalmente, o *Espaço de Contexto* possui as seguintes propriedades:

- Propagação Hierárquica do *Fluxo de Percepção* (Definição 2.2.1.6). Mesmo que *Símbolos* percebidos (e *Símbolos Compostos*) sejam propagados/distribuídos para múltiplas relações semânticas em múltiplos *Contextos* (por causa dos Relacionamentos Semânticos Multidimensionais, em Definição 2.3.1), o Teorema 2.2.1 prova que o *Fluxo de Percepção* propaga hierarquicamente dentro do *Espaço de Contexto* e esta propagação preserva as propriedades do *Framework SAIC* (Definição 2.1).
- Mudanças Incrementais. Uma **Mudança Incremental** no *Espaço de Contexto* é a mudança incremental, mútua e sincronizada em ambos o *Espaço de Percepção* (Definição 2.2.1.5) e o Modelo de Mundo (Definição 2.2.2.4). Basicamente, a *Mudança Incremental* significa que: a percepção incremental (do *Ambiente* do mundo real) causa transições incrementais (entre os estados dentro do Modelo de Mundo). O Teorema 2.2.2 prova que uma *Mudança Incremental* no *Espaço de Contexto* resulta da operação de Casamento de Sintaxe e Semântica (Definição 2.2.2.5). O Exemplo 2.2.5 (abaixo) ilustra alguns cenários de *Mudança Incremental*.

Exemplo 2.2.5 Sobre Mudanças Incrementais no Espaço de Contexto.

Cenário de Ambiente Virtual: Num *Ambiente* virtual planejado (p.ex. interfaces gráficas de aplicação hipermídia), o usuário realiza *Mudanças Incrementais* no *Espaço de Contexto*, porque cada transição de um lugar para outro é feita sequencialmente, passo-a-passo (isto é, clique-a-clique). Esta transição é conceitualmente rápida, embora o dispositivo físico que suporta o *Ambiente* virtual possa introduzir algum *overhead* ou atraso. Adicionalmente, estes lugares ligados estão semanticamente próximos no

Espaço de Contexto. Por exemplo, quando o usuário abre uma interface de uma aplicação, os *links* ou botões exibidos nesta interface – sempre que ativados – vão conduzir o usuário para um lugar cuja funcionalidade seja compatível com o propósito da aplicação.

Cenário de Ambiente Físico: Considerando a transição de um lugar para outro num *Ambiente* físico, uma pessoa realiza *Mudanças Incrementais* no *Espaço de Contexto*. Cada movimento de um lugar para outro envolve a aceleração da massa do corpo, e uma vez em movimento, a massa obedece a uma trajetória contínua. Portanto, se é contínua, então a mudança de *Contextos* no *Espaço de Contexto* é incremental. Aqui, a velocidade da transição (andando, dirigindo um carro, pilotando um avião ou espaçonave) de um lugar para outro é usualmente proporcional à distância física da trajetória. Portanto, em *Ambientes* físicos, quanto mais rápida é a velocidade de transição, maior o *Ambiente* físico deve ser – o que não implica que sua representação dentro de um Modelo de Mundo deva ser igualmente enorme.

Observação 1) Estes conceitos de *Espaço de Contexto* e *Mudanças Incrementais* são melhor explicados durante a descrição do Modelo de Mundo (Definição 2.2.2.4), o Casamento de Sintaxe e Semântica (Definição 2.2.2.5) e o Teorema 2.2.2.

Observação 2) A propriedade de *Mudanças Incrementais* é fundamental para os serviços incrementais do SADI. Além disso, o *Framework* SADI considera que vai operar (e inter-operar) com tecnologias de software que usam o *Framework* SAIC (p.ex. [ABO 99]), que significa basicamente serviços com operações incrementais (independentes da quantidade total dos dados) e serviços com crescimento contínuo.

Todavia, existem casos onde a propriedade de *Mudanças Incrementais* não ocorre continuamente. Em tais casos, a **Detecção de Contexto Massiva** é a operação que detecta *Contextos* depois de uma mudança abrupta no *Ambiente* percebido. Por exemplo, alguém desmaia e depois acorda num lugar diferente com nenhum rosto familiar ao redor. Esta pessoa realiza uma *Detecção de Contexto Massiva* enquanto pergunta: ‘onde estou’, ‘quem são vocês’, ‘o que aconteceu’, etc. Da mesma forma, um agente assistente de software realiza uma *Detecção de Contexto Massiva* enquanto o sistema operacional do computador está carregando a configuração personalizada do espaço de trabalho de uma nova sessão (*login*) de um usuário.

Teorema 2.2.1 A propagação do *Fluxo de Percepção* (Definição 2.2.1.6) dentro do *Espaço de Contexto* (Definição 2.2.2.3) é hierárquica, com comprimento de propagação no máximo $\log_2(N^2_Símbolos_percebidos)$, e esta propagação preserva as propriedades do *Framework* SAIC (Definição 2.1) – as quais são: serviços escaláveis, incrementais e contínuos.

PROVA:

São considerados dois **Casos**: **1)** Existe apenas um *Símbolo* na *Entrada de Percepção* (Definição 2.2.1.4); e, **2)** Existem dois ou mais *Símbolos* na *Entrada de Percepção*.

Caso 1: A *Entrada de Percepção* contém apenas um *Símbolo*.

Este único *Símbolo* é oriundo de apenas uma hierarquia dentro do *Espaço de Percepção* (Definição 2.2.1.5), isto é, ele é gerado por apenas um sensor (ou *parser*).

Portanto, dado que este único *Símbolo* não vai combinar com outro *Símbolo*, então o *Fluxo de Percepção* para este único *Símbolo* dentro do *Espaço de Contexto* é realizado usando a funcionalidade primitiva da Arquitetura de Serviços SAIC (Definição 2.1), isto é, o *Framework SAIC*.

Caso 2: A *Entrada de Percepção* contem dois ou mais *Símbolos*.

Neste 2º *Caso*, é considerado duas **Situações**: 1) Existe apenas um *Contexto*; e, 2) Existem dois ou mais *Contextos*.

Agora é demonstrado que os *Símbolos Compostos* (Definição 2.2.1.2) não alteram as propriedades escalável, incremental e contínua do *Framework SAIC*. A demonstração usa dois *Símbolos* *A* e *B* que formam um *Símbolo Composto* *AB* numa sub-árvore. A sub-árvore é usada aqui como o bloco de construção para a hierarquia inteira (assim como na prova do Teorema 2.1.1).

Situação 1) Primeiro, considere que o *Fluxo de Percepção* propaga dentro de apenas um *Contexto*, isto é, os *Símbolos* *A*, *B* e o resultante *Símbolo Composto* *A.B* não são replicados/propagados para outros *Contextos*.

Usando o *Framework SAIC*, mais precisamente: usando o *Serviço de Filtragem* (Definição 2.1.2), a percepção dos *Símbolos* *A* e *B* ao mesmo tempo faz cada um seguir uma sub-árvore distinta (ver Figura 3.7-A).

Usando o *Framework SADI*, mais precisamente: usando o *Serviço de Detecção de Contexto* (Definição 2.2.2), a percepção dos *Símbolos* *A* e *B* ao mesmo tempo também faz cada um seguir uma sub-árvore distinta. Todavia, estas sub-árvores sobrepõem seus *Símbolos Compostos* comuns (ver Figura 3.7-B).

Portanto, nesta *Situação 1* (*Contexto* único), o *Serviço de Detecção de Contexto* continua a operar com hierarquias, isto é, os *Símbolos Compostos* dentro do *Fluxo de Percepção* não alteram a propagação hierárquica.

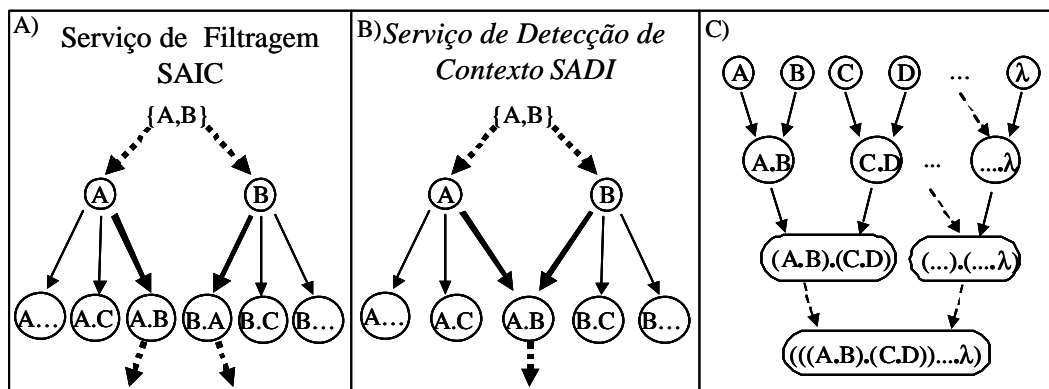


Figura 3.7: A) Serviço de Filtragem SAIC B) Serviço de Detecção de Contexto SADI C) A Propagação de Símbolos Compostos

Além disso, cada *Símbolo Composto* é formado por pelo menos dois *Símbolos Compostos* menores (ou pelo menos dois *Símbolos* únicos, no início). Assim, para cada nível de propagação de *Símbolos Compostos*, o conjunto de *Símbolos Compostos* que propaga é dividido por dois (ver Figura 3.7-C). Portanto, o comprimento de propagação do *Fluxo de Percepção* é $O(\log_2(N^2_{\text{Símbolos_percebidos}}))$.

Situação 2) Agora, considere que o *Fluxo de Percepção* propaga em dois ou mais *Contextos*, isto é, os *Símbolos A, B* e o resultante *Símbolo Composto A.B* são replicados/propagados para outros *Contextos*.

As condições da *Situação 2* são quase as mesmas da *Situação 1*. Conforme mostra a Figura 3.8, a única diferença entre a *Situação 2* e a *Situação 1* é: a replicação/propagação dos *Símbolos A* e *B* para outros *Contextos* (em t_{i+1}); e, a replicação/propagação do *Símbolo Composto A.B* para outros *Contextos* (em t_{i+3}).

Apesar da replicação/propagação de *Símbolos* e *Símbolos Compostos* (respectivamente em t_{i+1} e t_{i+3}), o *Serviço de Detecção de Contexto* continua a operar em hierarquias (as operações em t_i , t_{i+2} e t_{i+4} , mostradas na Figura 3.8-B). Portanto, os *Símbolos Compostos* dentro do *Fluxo de Percepção* (através de múltiplos *Contextos*) não alteram a propagação hierárquica. Aqui, o comprimento de propagação do *Fluxo de Percepção* também é $O(\log_2(N^{\text{º}}_{\text{Símbolos_percebidos}}))$.

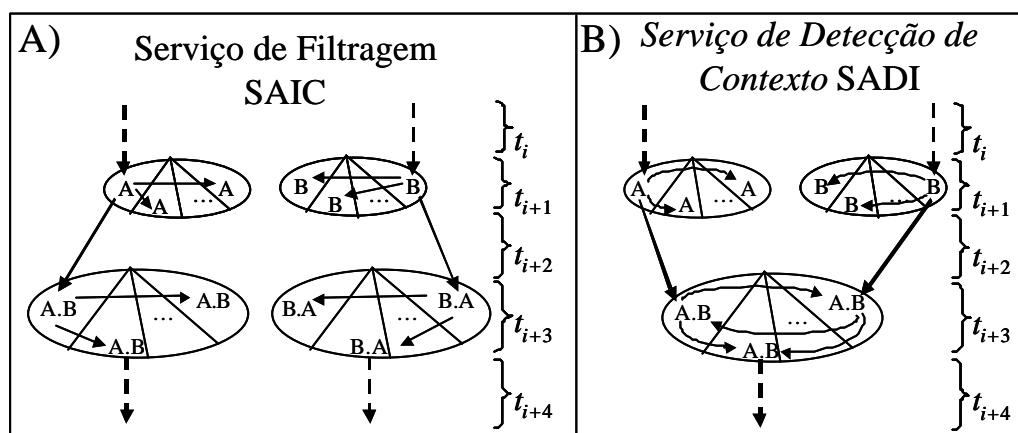


Figura 3.8: A) Serviço de Filtragem SAIC B) Serviço de Detecção de Contexto SADI

Definição 2.2.2.4 Modelo de Mundo

O **Modelo de Mundo** é um modelo conceitual Orientado a Objeto do *Ambiente* (Definição 1.3, e estendida nas Definições 2.2.1, 2.2.2 e 2.2.2.3). O *Framework* SADI assume que os sistemas de aplicação descrevem o *Modelo de Mundo*, o qual é baseado em Ontologias e *Frameworks* de Objetos.

O *Modelo de Mundo* fornece semântica para o *Espaço de Contexto* (Definição 2.2.2.3) e ele também pode ser usado para Planejamento (em Definição 2.2.3).

O *Modelo de Mundo* descreve os Objetos do *Ambiente*, as Ações a serem realizadas sobre (ou relacionadas com) os Objetos do *Ambiente*, e as funções de recompensa (*reward*) do *Feedback Interior*. O **Feedback Interior** é uma subclasse especial dos Objetos do *Ambiente* e ele fornece orientação sobre quão bom/certo/sucesso (ou quão ruim/errado/fracasso) é o resultado/efeito de uma Ação realizada sobre (ou relacionada com) um Objeto do *Ambiente*. Note que o *Feedback Interior* é relativo à cada *Contexto* (Definição 2.2.2.1).

No *Framework* SADI, o *Modelo de Mundo* possui um relacionamento semântico importante com o *Espaço de Percepção* (Definição 2.2.1.5). Por esta razão, a descrição do *Modelo de Mundo* deve ter:

- Um conjunto finito de estados, onde cada estado é chamado **Estado de Ação A_k** . Um A_k pode ter sub-estados, onde cada sub-estado é também um *Estado de Ação* (a notação para sub-estado é $A_{k.i}$);
- Um conjunto finito de gatilhos (*triggers*), onde cada gatilho é chamado **Símbolo-Disparador $S_k:?$** . Um $S_k:?$ é um *Símbolo* especial que está associado com um A_k (ou $A_{k.i}$). O $S_k:?$ deve ser casado dinamicamente (pelo Casamento de Sintaxe e Semântica (Definição 2.2.2.5)) com um *Símbolo* S_p do *Espaço de Percepção* (isto é, o '?' de $S_k:?$ é substituído por S_p). Este casamento dinâmico permite um A_k fazer uma transição para outro A_j (isto é, $S_k:S_p @ A_j$) ou para um de seus sub-estados $A_{k.i}$ (isto é, $S_k:S_p @ A_{k.i}$);
- Um conjunto finito de relacionamentos semânticos entre o *Modelo de Mundo* e o *Espaço de Percepção*, onde cada relacionamento é chamado **Relacionamento-Transição $S_k:?-S_k$** . Um *Relacionamento-Transição* ($S_k:?-S_k$) liga a semântica do *Símbolo* S_k (de $S_k:?$ do *Modelo de Mundo*) à sintaxe do *Símbolo* S_k (do *Espaço de Percepção*). O $S_k:?-S_k$ fornece o ponto de entrada no *Espaço de Percepção* onde o Casamento de Sintaxe e Semântica deve iniciar.

Observação) Estes conjuntos finitos de *Estados de Ação* (A_k), *Símbolos-Disparadores* ($S_k:?$) e *Relacionamentos-Transição* ($S_k:?-S_k$) podem ser considerados como regras de crescimento dinâmico, guiando o crescimento da Representação de Conhecimento SADI (Definição 2.3), durante a interação dos sistemas de aplicação (baseados no SADI) com o *Ambiente* complexo, dinâmico e incerto. Note que o ciclo de operação contínuo do SADI funciona neste *Ambiente* como uma Máquina de *Feedback* (*Feedback Machine* na aceção da Teoria do Caos).

Definição 2.2.2.5 Casamento de Sintaxe e Semântica

Durante a contagem lógica do *Tempo de Sistema* T_S (Definição 2.2.1.3), o *Serviço de Detecção de Contexto* (Definição 2.2.2) recebe do *Serviço de Percepção do Ambiente* (Definição 2.2.1) a *Entrada de Percepção* (Definição 2.2.1.4) que é também usada para alimentar adiante (*feedforward*) a ativação de *Contextos* (Definição 2.2.2.1).

Ainda em T_S , um *Contexto* é ativado apenas quando: 1^o) ele recebe o *Fluxo de Percepção* (Definição 2.2.1.6); e, 2^o) a semântica de suas *Percepções Incertas* (dentro do *Fluxo de Percepção*) são definidas dinamicamente pela operação de **Casamento de Sintaxe e Semântica**, que funciona da seguinte forma:

Passo 1. Em T_S , é lido o conjunto S_{S-D} de *Símbolos-Disparadores* ($S_k:?$) do atual *Estado de Ação* (A_k) ativo do *Modelo de Mundo* (Definição 2.2.2.4);

Passo 2. Para cada $S_k:?$ no conjunto S_{S-D} :

2.1 Realiza uma busca_em_profundidade (em T_S), começando no ponto de entrada indicado pelo *Relacionamento-Transição* ($S_k:?-S_k$);

Observação 2.1.1) A busca_em_profundidade segue a orientação da propagação hierárquica do *Fluxo de Percepção* (em T_S);

Observação 2.1.2) A busca_em_profundidade termina quando:

1^o) Ela alcança o maior *Símbolo Composto* que resultou do *Fluxo de Percepção* (em T_S); ou

2º) Ela alcança o limite do **Comprimento de Busca**. Este parâmetro de comprimento é pré-configurado pelo *Framework SADI*, mas pode ser reconfigurado pelas aplicações baseadas no SADI;

Observação 2.1.3) Durante a busca_em_profundidade em T_S , um **Coefficiente de Casamento** de $1/\alpha$ ($\alpha = 1, 2, \dots, n$) determina o quanto um Símbolo Composto incompleto (alcançado pela busca_em_profundidade) pode ser completado com seus componentes mais prováveis (isto é, os menores *Símbolos Compostos* de sua vizinhança).

Todavia, este Símbolo Composto incompleto deve ter recebido pelo menos $1-1/\alpha$ de seus componentes (se $\alpha = 1$, a busca_em_profundidade apenas aceita *Símbolos Compostos* completos).

O *Coefficiente de Casamento* é também pré-configurado pelo *Framework SADI*, mas pode ser reconfigurado pelas aplicações baseadas no SADI;

Observação 2.1.4) Estas buscas_em_profundidade (no conjunto S_{S-D} , do Passo 2) podem ser realizadas em paralelo (em T_S);

Passo 3. Se o conjunto de *Símbolos Compostos* (retornado do Passo 2) não for vazio, então retorna (em T_S) este conjunto. No caso de ser um conjunto vazio, é retornado um **Vazio_Status** para as aplicações baseadas no SADI.

Observação)

O seguinte Exemplo 2.2.6 ilustra a operação do *Casamento de Sintaxe e Semântica*. Depois, o Teorema 2.2.2 prova que uma *Mudança Incremental no Espaço de Contexto* resulta da operação de *Casamento de Sintaxe e Semântica*.

Exemplo 2.2.6 Sobre o *Casamento de Sintaxe e Semântica*

Um andróide decide caminhar. Inicialmente, em T_S , o objetivo de caminhar do andróide define o *Contexto* 'Movimento', ao ativar o correspondente *Estado de Ação* 'Movimento' no seu *Modelo de Mundo* (ver Figura 3.9-A) – onde o *Estado de Ação* 'Mover' é o sub-estado inicial *default*. (Devido à simplicidade deste exemplo, os parâmetros *Coefficiente de Casamento* e *Comprimento de Busca* não são usados durante a operação).

Dado o *Fluxo de Percepção* (em T_S) ilustrado na Figura 3.9-B, a operação de *Casamento de Sintaxe e Semântica* produz os seguintes resultados:

Passo 1:

$A_k = \text{'Move'}$;
 $S_{S-D} = \{\text{'Limpo:?'}, \text{'Obstáculo:?'}\}$;

Passo 2:

A busca_em_profundidade iniciada em (Limpo:?– Limpo) retorna 'Limpo.Terreno';

A busca_em_profundidade iniciada em (Obstáculo:?–Obstáculo) retorna um conjunto vazio.

Passo 3:

'Limpo.Terreno' é retornado.

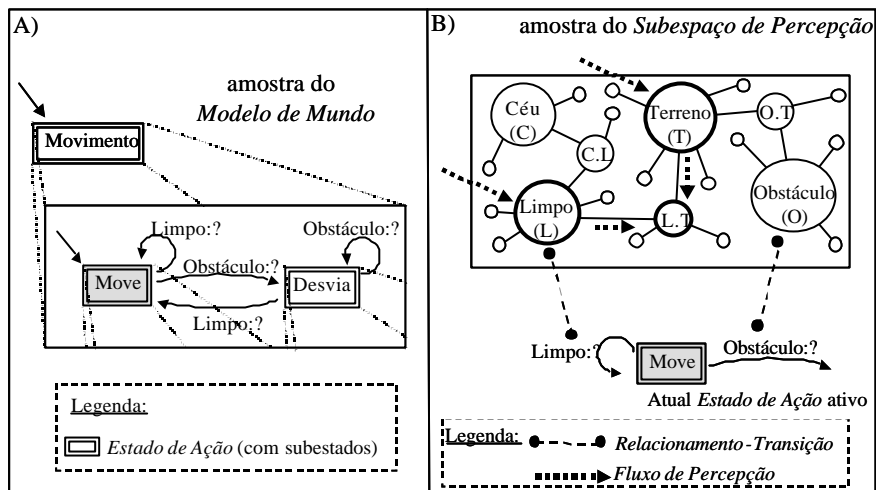


Figura 3.9: A) Exemplo de Modelo de Mundo; B) Fluxo de Percepção e relacionamentos entre o Modelo de Mundo e o Espaço de Percepção

Note que, o atual *Estado de Ação* ativo (em T_S) do *Modelo de Mundo* e os *Símbolos-Disparadores* associados com ele, eles atuam como atratores neste sistema dinâmico onde os *Símbolos* percebidos (em T_S) têm suas semânticas definidas. Por esta razão, no Exemplo 2.2.6, a percepção de 'Terreno' é imediatamente associada com o *Símbolo-Disparador* 'Limpido:?' do atual *Estado de Ação* ativo 'Mover'. Na Figura 3.10, este cenário é ilustrado na perspectiva dos três principais subespaços (PO, PA, PI) do *Espaço de Percepção*. A construção destes subespaços é melhor descrita no Serviço de Tomada de Decisão (Definição 2.2.3) e Serviço de Aprendizagem (Definição 2.2.4).

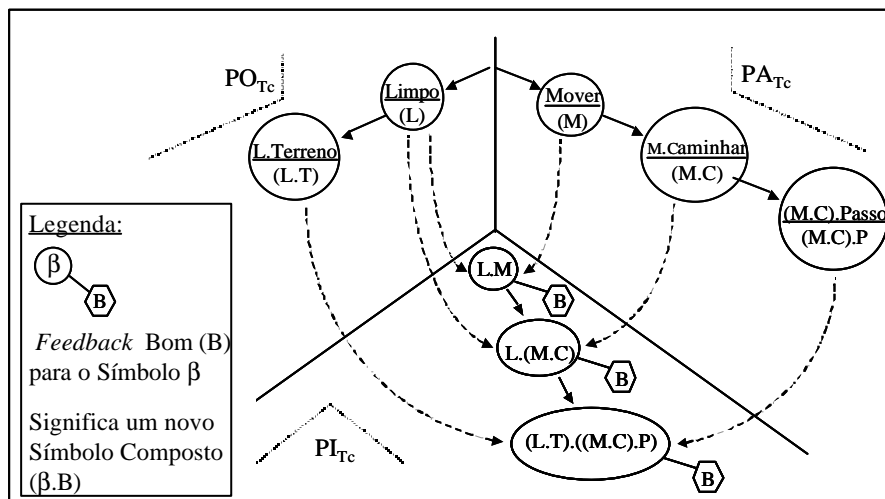


Figura 3.10: Exemplo dos subespaços PO, PA, PI

Teorema 2.2.2 Dado que a *Mudança Incremental* (em T_S) no *Espaço de Contexto* (Definição 2.2.2.3) é a mudança (em T_S) incremental, mútua e sincronizada em ambos o *Espaço de Percepção* (Definição 2.2.1.5) e o *Modelo de Mundo* (Definição 2.2.2.4), então a *Mudança Incremental* resulta do *Casamento de Sintaxe e Semântica* (em Definição 2.2.2.5).

PROVA:

Dado que:

O *Espaço de Contexto* (em Definição 2.2.2.3) contém a união da sintaxe (do *Espaço de Percepção*, Definição 2.2.1.5) e da semântica (do *Modelo de Mundo*, Definição 2.2.2.4);

Adicionalmente, cada mudança incremental (em T_S) no *Ambiente* gera um *Fluxo de Percepção* (com *Símbolos* e *Símbolos Compostos*) que propagam de forma incremental e hierárquica (Teorema 2.2.1) através de múltiplos *Contextos* (em T_S);

Portanto:

Um conjunto de *Contextos* recebe o *Fluxo de Percepção* (em T_S), mas um *Contexto* é apenas ativado (em T_S) se ele também tem a semântica de suas Percepções Incertas (dentro deste *Fluxo de Percepção*) dinamicamente definida (em T_S) pelo *Casamento de Sintaxe e Semântica*;

Ainda considerando que:

O *Casamento de Sintaxe e Semântica* realiza uma busca_em_profundidade (em Definição 2.2.2.5), em T_S , para cada *Símbolo-Disparador* $S_k:?$ do atual *Estado de Ação* A_k ativo do *Modelo de Mundo* (em T_S);

Portanto:

O atual Estado de Ação A_k ativo faz (em T_S) uma transição (para outro A_j ou $A_{k.i}$) se e somente se um de seus Símbolos-Disparadores $S_k:?$ casa (em T_S) com um *Símbolo* S_p do *Fluxo de Percepção* dentro do *Espaço de Percepção*;

Se ocorrem (em T_S) transições incrementais entre *Estados de Ação* para cada *Fluxo de Percepção* incremental (em T_S) (porque as transições entre *Estados de Ação* são guardadas por *Símbolos-Disparadores*), então o *Modelo de Mundo* também faz uma mudança incremental, e conseqüentemente o *Espaço de Contexto* (que contém a união da sintaxe do *Espaço de Percepção* com a semântica do *Modelo de Mundo*).

Esta mudança sincronizada e incremental (em T_S) em ambos o *Espaço de Percepção* e o *Modelo de Mundo* é também mútua (em T_S) porque:

O atual Estado de Ação ativo A_k do *Modelo de Mundo* (em T_S , com seus Símbolos-Disparadores $S_k:?$) influencia (Passo 1, em Definição 2.2.2.5) a semântica das *Percepções Incertas* dentro do *Fluxo de Percepção* (em T_S);

Mas:

Uma vez que as Percepções Incertas recebem suas semânticas, isto é, quando elas são alcançadas pela busca_em_profundidade e os $S_k:S_p$ casados são retornados (Passos 2 e 3, em Definição 2.2.2.5), portanto elas influenciam as transições incrementais entre *Estados de Ação* do *Modelo de Mundo*, isto é, $S_k:S_p \textcircled{R} A_j$ ou $S_k:S_p \textcircled{R} A_{k.i}$.

Definição 2.2.2.6 Contexto Estendido

Dado que o *Espaço de Contexto* (Definição 2.2.2.3) contem o modelo conceitual que guia o crescimento da Representação de Conhecimento SADI (Definição 2.3), um **Contexto Estendido** é uma instância de um *Contexto* (Definição 2.2.2.1) definido no *Espaço de Contexto*.

Quando um *Contexto* é ativado (ao receber o *Fluxo de Percepção* e a definição dinâmica da semântica de suas *Percepções Incertas*), então um respectivo *Contexto Estendido* é instanciado para receber os *Símbolos* (e *Símbolos Compostos*) reconhecidos neste *Contexto* ativado.

Um *Contexto Estendido* recebe as instâncias dos *Símbolos* (e *Símbolos Compostos*) reconhecidos e então organiza/classifica estes *Símbolos* em **Classes que Correspondem às Estruturas de Linguagem (CCEL)**, as quais possuem a seguinte estrutura:

- Sentença:
 - Sujeito: Quem Q realizou a ação A ;
 - Predicado:
 - Verbo: Qual é a ação A ;
 - Objeto (e Adjetivos): Qual é o objeto O relacionado com a ação A ;
 - Advérbios:
 - Onde o conjunto $\{Q, A, O\}$ ocorreu;
 - Quando o conjunto $\{Q, A, O\}$ ocorreu;
 - Como o conjunto $\{Q, A, O\}$ ocorreu;
 - Por quê o conjunto $\{Q, A, O\}$ ocorreu;

As classes *CCEL* (Sujeito, Verbo, Objeto, Onde, Quando, Como, Por quê) representam subespaços na Representação de Conhecimento SADI (Definição 2.3), onde cada classe *CCEL* contem uma hierarquia de instâncias.

A organização/classificação de instâncias dentro das classes *CCEL* são realizadas usando a funcionalidade primitiva do *Framework* SAIC (Definição 2.1), porque as hierarquias de classes *CCEL* não se sobrepõem (porque elas contem instâncias de *Símbolos Compostos* já formados e reconhecidos nos Contextos ativados). Todavia, existe uma adaptação: numa dada sub-árvore de uma hierarquia, se a instância recebida não casar com nenhum *Contexto Estendido*, isto é, a similaridade entre a instância recebida e o *Contexto Estendido* não é maior que um certo limiar, então uma nova ramificação é criada nesta sub-árvore (ver Exemplo 2.2.7).

Observação 1) É importante distinguir entre: os subespaços do *Espaço de Percepção* (que representam apenas os *Símbolos* percebidos pelos sensores ou *parsers*); e, os subespaços das classes *CCEL* (que representam instâncias dos *Símbolos* percebidos e reconhecidos nos *Contextos* ativados). Por exemplo, Verbo e Objeto são respectivamente as classes *CCEL* para *PA* e *PO* (em Definição 2.2.1.5). Na Figura 3.11, o cenário do Exemplo 2.2.6 é ilustrado na perspectiva das classes *CCEL* (onde, para simplificar a figura, nem todas as classes *CCEL* são mostradas combinadas dentro do subespaço PI_{Tc}).

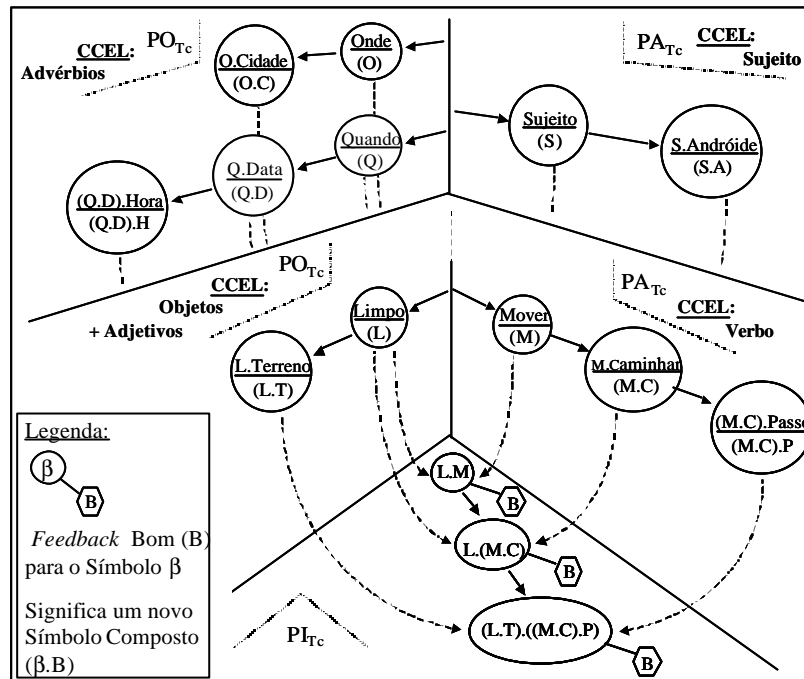


Figura 3.11: Exemplo de uma instância de classe *CCEL*

Observação 2) Apesar da definição das classes *CCEL*, a descrição restante do *Framework SADI* continua a focalizar os conceitos de *PO*, *PA* e *PI* (em Definição 2.2.1.5), porque eles possuem um relacionamento especial (que dá suporte às operações incrementais do *Framework SADI*) e também porque eles geram o cubo básico conceitual da Representação de Conhecimento SADI (Definição 2.3.2). A partir de agora, para simplificar a notação, *PA* conduz a {Sujeito, Verbo} e *PO* conduz a {Objeto, Onde, Quando, Como}.

O Por quê está implicitamente representado na operação de *Casamento de Sintaxe e Semântica* (Definição 2.2.2.5). Basicamente, o Por quê tem sua origem no atual *Estado de Ação* A_k ativo que contribuiu para a ativação do *Contexto* (que instanciou o *Contexto Estendido* correspondente). O Por quê pode ser interpretado sob o conceito de Intencionalidade [SEA 95], se for assumida uma analogia com Estados Intencionais [ibid]: o Conteúdo Intencional de um Estado Intencional determina as Condições de Satisfação que devem ser alcançadas para tornar este Estado satisfeito. No Exemplo 2.2.6, o *Estado de Ação* 'Mover' é o "Estado Intencional", e o *Símbolo-Disparador* 'Limpo:?' é a "Condição de Satisfação a ser alcançada".

Exemplo 2.2.7 Com relação à organização/classificação de instâncias dentro das classes *CCEL*, se o andróide (representado na Figura 3.11 por '(S.A)') nadar ('Mover.Nadar' = '(M.N)') num rio ('Objeto.Rio' = '(O.R)'), então a instância '((S.A).(M.N)).(O.R))' é primeiramente roteada-por-conteúdo para o *Contexto Estendido* 'Movimento', onde '(S.A)' é roteada-por-conteúdo para a sub-árvore *Sujeito*, '(M.N)' para a sub-árvore *Verbo*, e '(O.R)' para a sub-árvore *Objeto*. Note que o único casamento ocorreu com o *Sujeito* '(S.A)', portanto novas ramificações são criadas para '(M.N)' e '(O.R)'. Apesar do roteamento-por-conteúdo dos componentes da instância {'(S.A)', '(M.N)', '(O.R)'}, eles estão todos ligados juntos com a estrutura de dados que representa a instância inteira no subespaço de *Percepção de Feedback Interior* (PI_{Tc}).

Definição 2.2.3 Serviço de Tomada de Decisão

Considere o ciclo de operação do *Framework* SADI (isto é, o ciclo: Percepção do Ambiente, Detecção de Contexto, Aprendizagem e Tomada de Decisão), que corresponde a um *Tempo de Sistema* T_S (Definição 2.2.1.3).

O **Serviço de Tomada de Decisão** é um processo de decisão incremental, *top-down* e adaptativo realizado toda vez que um conjunto de *Símbolos-Disparadores* casados $S_k:S_p$ é retornado (em T_S) pelo *Casamento de Sintaxe e Semântica* (Definição 2.2.2.5) do *Serviço de Detecção de Contexto* (Definição 2.2.2).

Em T_S , o *Serviço de Tomada de Decisão* é realizado depois do Serviço de Aprendizagem (Definição 2.2.2.4), porque o *Serviço de Tomada de Decisão* necessita trabalhar com o conhecimento mais recente que o Serviço de Aprendizagem atualiza no *Contexto Estendido* ativado (Definição 2.2.2.6) da Representação de Conhecimento SADI (Definição 2.3).

O *Serviço de Tomada de Decisão* trabalha com Caminhos de Decisão. Um **Caminho de Decisão** é uma seqüência de transições entre *Estados de Ação* A_k (dentro dos *Frameworks* de Objetos do *Modelo de Mundo*, Definição 2.2.2.4), onde:

- A seqüência inicia num A_k que é um dos **Estado de Ação Mais Abstrato (A_{+k})** (devido à abordagem *top-down* do *Serviço de Tomada de Decisão*);
- A seqüência continua com transições que podem ser de um A_k para:
 - Um de seus sub-estados $A_{k,i}$ (onde $A_{k,i}$ pertence à mesma Classe de Objeto);
 - Outro estado A_j (onde A_j pertence a outra Classe de Objeto);
- A seqüência finaliza num A_k que é um dos **Estado de Ação Menos Abstrato (A_{-k})**. Este A_{-k} pode estar:
 - Na mesma Classe de Objeto, se apenas ocorreram transições tipo $S_k:S_p \textcircled{R} A_{k,i}$, ou um circuito de transições ($S_k:S_{p1} \textcircled{R} A_j, S_j:S_{p2} \textcircled{R} \dots, \dots, S_{pn} \textcircled{R} A_{k,i}$); ou,
 - Em outra Classe de Objeto, se ocorreu uma transição do tipo $S_k:S_p \textcircled{R} A_j$, e nenhum circuito de transições é formado de volta a algum $A_{k,i}$.

Cada *Estado de Ação* A_k (de um *Caminho de Decisão*), juntamente com seu respectivo conjunto S_{S-D} de *Símbolos-Disparadores* ($S_k:?$) formam uma **Célula de Decisão**. Para cada *Célula de Decisão*, o *Casamento de Sintaxe e Semântica* para o conjunto S_{S-D} pode gerar três situações possíveis:

- 1^ª) Nenhum $S_k:?$ é casado e o *Comprimento de Busca* (Passo 2, Definição 2.2.2.5) é atingido. Neste caso, a aplicação baseada no SADI recebe um *Vazio_Status* (Passo 3, Definição 2.2.2.5). Adicionalmente, o *Serviço de Detecção de Contexto* indica o *Estado de Ação* A_p mais provável do *Modelo de Mundo* (onde $A_p \neq A_k$), dada a atual *Entrada de Percepção* (Definição 2.2.1.4) em T_S . Neste caso, o atual *Estado de Ação* A_k ativo do *Modelo de Mundo* é modificado para se adaptar à realidade externa do Ambiente do mundo real;
- 2^ª) Apenas um $S_k:?$ é casado com outro *Símbolo* S_p (isto é, $S_k:S_p$), e então o *Modelo de Mundo* muda seu atual *Estado de Ação* ativo de A_k para $A_{k,i}$ (via $S_k:S_p \textcircled{R} A_{k,i}$) ou de A_k para A_j (via $S_k:S_p \textcircled{R} A_j$);
- 3^ª) Mais de um $S_k:?$ são casados, isto é, existe um conjunto $\{S_{k1}:S_{p1}, S_{k2}:S_{p2}, \dots, S_{kn}:S_{pn}\}$. Agora o *Serviço de Tomada de Decisão* usa a Representação de Conhecimento SADI (Definição 2.3), para selecionar o $A_{k,i}$ (ou A_j) que possui a maior probabilidade de uma **Percepção de Feedback Interior Bom/certo/sucesso**

(F_B), dado o **Símbolo Casado** $S_k:S_p$ (S_C) e o atual *Estado de Ação* A_k ativo. Então, para cada candidato A_c ($A_c = A_{k,i}$ ou $A_c = A_j$) é calculado:

$$P(F_B / S_C, A_c) = P(F_B, S_C, A_c) / P(S_C, A_c)$$

Adicionalmente, a aplicação baseada no SADI pode pesquisar na Representação de Conhecimento SADI pela probabilidade de uma **Percepção de Feedback Interior Ruim/errado/insucesso** (F_R), para compará-la com F_B . Da mesma forma, para cada candidato A_c é calculado:

$$P(F_R / S_C, A_c) = P(F_R, S_C, A_c) / P(S_C, A_c)$$

Na Representação de Conhecimento SADI, é descrito o que acontece toda vez que uma aplicação baseada no SADI requisita uma probabilidade condicional $P(F_B, S_C, A_c)$, $P(F_R, S_C, A_c)$ ou $P(S_C, A_c)$ que não está representada num dado *Contexto Estendido* (Definição 2.2.2.6). Basicamente, o *Framework* SADI realiza um processo de aproximação usando *Símbolos* alternativos (porém, relacionados). Se nenhum *Símbolo* alternativo é encontrado, então a aplicação baseada no SADI recebe um *Não_Encontrado_Status* e então decide como lidar com esta exceção (p.ex. usando uma ação *default*, ou escolhendo uma ação candidata randomicamente, etc).

Exemplo 2.2.8 Sobre o *Serviço de Tomada de Decisão*

A Figura 3.12 ilustra o *Serviço de Tomada de Decisão* incremental e adaptativo, ao expandir o cenário do Exemplo 2.2.6. Note que, na Figura 3.12, $S_k:\emptyset @ A_{ki}$ significa um *Símbolo-Disparador* que não participa de um *Símbolo Composto* para aquele *Contexto* (isto é, apenas o *Símbolo* 'Próprio_corpo' é necessário para disparar o *Estado de Ação* 'Andar').

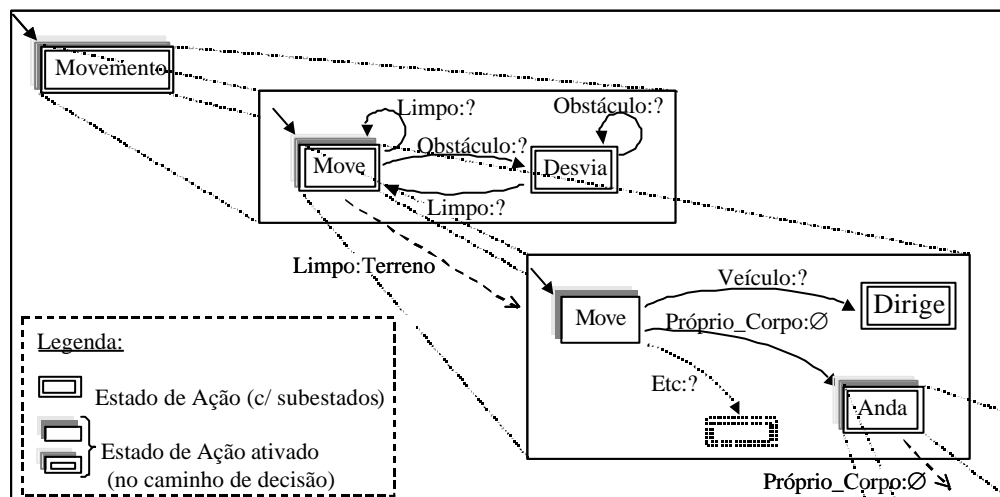


Figura 3.12: Exemplo de Decisão Incremental

Comentários sobre Planejamento

Cada decisão incremental no *Serviço de Tomada de Decisão* é definida como uma **Ação de Decisão**. Uma *Ação de Decisão* é conceitualmente uma **Ação de Planejamento** com uma **Janela Look-ahead** de tamanho zero, isto é, o *Ato de Decisão* usa (em T_S) apenas o atual *Estado de Ação* A_k ativo do *Modelo de Mundo* (com seu conjunto S_{S-D} de *Símbolos-Disparadores* $S_k:?$).

Todavia, o *Ato de Planejamento* estende a *Janela Look-ahead* ao realizar (em T_S) a simulação de eventos futuros (que podem provavelmente ocorrer em T_{S+t} , $t = 1, 2, \dots, n$).

O *Ato de Planejamento* constrói um plano composto de **Caminhos de Decisão Simulados**, onde um *Caminho de Decisão Simulado* inicia num dado *Estado de Ação* A_k e então cria uma seqüência de **Percepção de Objetos do Ambiente Simulada (POS_{T_c})**, **Percepção de Ação Simulada (PAS_{T_c})**, e **Percepção de Feedback Interior Simulada (PIS_{T_{c+1}})**.

O *Ato de Planejamento* usa a Representação de Conhecimento SADI (Definição 2.3) e o *Modelo de Mundo* (Definição 2.2.2.4) para realizar as simulações. Todavia, os $PIS_{T_{c+1}}$ não são introduzidos na Representação de Conhecimento SADI (isto é, eles são mantidos numa estrutura de dados temporária que é removida quando o *Ato de Planejamento* é concluído).

Enquanto um *Ato de Decisão* inicia num *Estado de Ação Mais Abstrato* (A_{+k}) (dos *Frameworks* de Objetos do *Modelo de Mundo*) e finaliza num *Estado de Ação Menos Abstrato* (A_{-k}), o *Ato de Planejamento* pode iniciar em qualquer *Estado de Ação* A_k (em qualquer nível de abstração). Além disso, o *Ato de Planejamento* cria *Caminhos de Decisão Simulados* usando a abordagem *top-down* que opera primeiramente nos *Estados de Ação* cujo nível de abstração é o mesmo do *Estado de Ação* A_k inicial. Esta operação dentro do mesmo nível de abstração permite a escolha de *Estados de Ação* que fornecem um maior Valor (de acordo com a Função de Valor definida na Aprendizagem por Reforço, [SUT 99]), porque os *Estados de Ação* (de maior nível de abstração) tem acesso às informações generalizadas (de *Percepções de Feedback Interior*) que são retro-propagadas (*backpropagated*) de outros *Estados de Ação* (de menor nível de abstração) – conforme descrito no Serviço de Aprendizagem (Definição 2.2.4).

Quando o *Ato de Planejamento* cria um *Caminho de Decisão Simulado* dentro do mesmo nível do *Estado de Ação* A_k inicial, o *Ato de Planejamento* pode melhorar/refinar o plano fazendo operações recursivas nos *Sub-Estados de Ação* $A_{k,i}$ de menores níveis de abstração. Quando o plano é concluído, o *Ato de Planejamento* estabelece/configura o *Símbolo-Disparador* que vai disparar o primeiro *Estado de Ação* do plano gerado.

O *Ato de Planejamento* vai tentar aprimorar/refinar o plano o máximo possível dentro da fração de tempo (em T_S) que está disponível para seu planejamento, mas deve ser considerado (pelas aplicações baseadas no SADI) que planos muito complexos e detalhados são prováveis de necessitar adaptações posteriores (em T_{S+t} , $t = 1, 2, \dots, n$) porque o *Ambiente* é dinâmico, complexo e incerto.

Definição 2.2.4 Serviço de Aprendizagem

Dado que:

- As operações incrementais do *Framework* SADI são realizadas no seguinte ciclo: Percepção do Ambiente, Detecção de Contexto, Aprendizagem e Tomada de Decisão;
- O conhecimento é organizado em *Contextos* (Definição 2.2.2.1) dentro de um *Espaço de Contexto* (Definição 2.2.2.3);
- As *Percepções de Objetos do Ambiente (PO)*, *Percepções de Ações (PA)* e *Percepções de Feedback Interior (PI)* (em Definição 2.2.1.5, em Observação 2 da Definição 2.2.2.6) possuem relacionamentos de causa-e-efeito;
- A *Entrada de Percepção* (em T_S) é usada simultaneamente para:

- Alimentar adiante (*Feedforward*) a ativação de um ou mais *Contextos*. Onde, cada *Contexto* ativado incrementa seu T_C , e ações realizadas em T_C podem causar possíveis efeitos/resultados (a serem percebidos na próxima ativação do *Contexto*, isto é, em T_{C+1});
- Realimentar (*Feedback*), para cada *Contexto* ativado (em T_C), os possíveis efeitos/resultados (das ações realizadas na ativação anterior do *Contexto*, isto é, em T_{C-1});
- Um *Contexto* é ativado se somente se: 1^ª) ele recebe o *Fluxo de Percepção* (Definição 2.2.1.6); e, 2^ª) suas *Percepções Incertas* são definidas dinamicamente pelo *Casamento de Sintaxe e Semântica* (Definição 2.2.2.5).

O Serviço de Aprendizagem

- Opera (em T_C) num *Contexto* ativado (e seu respectivo *Contexto Estendido*, Definição 2.2.2.6), imediatamente depois da operação do *Serviço de Detecção de Contexto* (Definição 2.2.2); e,
- Atualiza (em T_C) o conhecimento do *Contexto* ativado e seu respectivo *Contexto Estendido* dentro da Representação de Conhecimento SADI (Definição 2.3). Este conhecimento atualizado vai ser usado (em T_C) pelo *Serviço de Tomada de Decisão* (Definição 2.2.3).

O *Serviço de Aprendizagem* atualiza este conhecimento usando o controle semântico (fornecido pelo *Serviço de Detecção de Contexto*) e a relação de causa-e-efeito entre:

- O **Fluxo de Feedforward** (em T_C), isto é, o *Fluxo de Percepção* (Definição 2.2.1.6) que contem as *PO* e *PA*; e,
- O **Fluxo de Feedback** (em T_{C+1}), isto é, o *Fluxo de Percepção* que contem as *PO*, *PA* e *PI*.

Cada *Símbolo S* alcançado pelo *Fluxo de Percepção* (*Fluxo de Feedforward* ou *Fluxo de Feedback*) incrementa seu **Contador de Frequência Relativa** (*S.Contador*) que é usado como uma medida de probabilidade (conforme descrito na Representação de Conhecimento SADI, Definição 2.3).

Assim, nos *Contextos* alcançados por um *Fluxo de Percepção*, cada *Símbolo S* alcançado (por este fluxo) incrementa seu *S.Contador*. Estes *Símbolos* que são atualizados apenas no *Espaço de Contexto* são usados para gerar aproximações alternativas de valores de probabilidades, os quais são descritos nos Relacionamentos Causais (Definição 2.3.1).

Todavia, apenas no *Contexto Estendido* (de um *Contexto* ativado), o mais recente *Fluxo de Feedforward* em T_C forma *Símbolos Compostos* ($PO_{T_C}.PA_{T_C}$) que ficam esperando pelo *Fluxo de Feedback* em T_{C+1} . Na Figura 3.13-A, no plano que representa a percepção da ação (PA_{T_C}), está ilustrada a percepção do estado de ação inicial apontando para seus respectivos subestados de ação. Analogamente, no plano que representa a percepção do objeto (PO_{T_C}), está ilustrada a percepção do objeto inicial (que corresponde ao *Símbolo-Disparador*) apontando para seus respectivos objetos compostos (formados pelo fluxo de percepção). Ambos os símbolos de PA_{T_C} e PO_{T_C} , na medida em que são percebidos, formam no plano que representa a percepção do *feedback* interior, os ($PO_{T_C}.PA_{T_C}$) que ficam esperando pelo *Fluxo de Feedback* em T_{C+1} (as setas “FB”, na Figura 3.13-B), os quais retro-propagam (*backpropagate*) os ($PO_{T_C}.PA_{T_C}$). $PI_{T_{C+1}}$) em direção aos níveis de abstração maior na hierarquia dentro do subespaço/dimensão *PI*. Esta retro-progagação fornece um processo incremental de generalização, e uma generalização eficiente de experiências passadas significa uma operação eficiente em problemas com

conjuntos muito grandes de estados' [SUT 99]. Estes *Símbolos* que são atualizados no *Contexto Estendido* são usados para fornecer valores exatos e também valores aproximados de probabilidades, os quais são descritos nos Relacionamentos Causais (Definição 2.3.1).

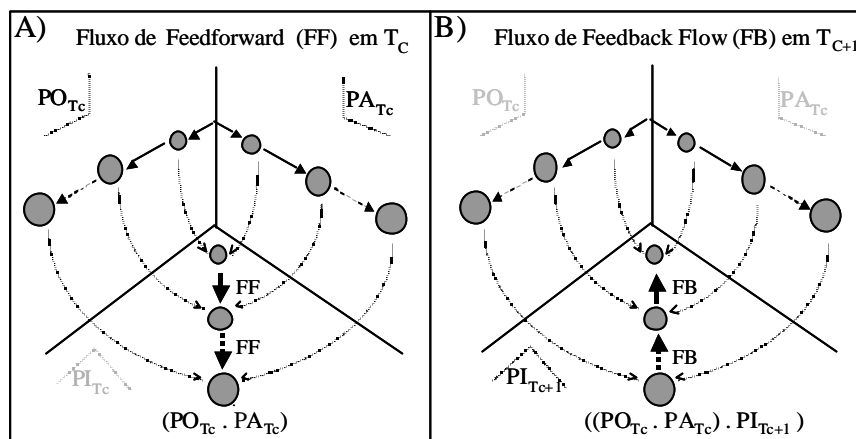


Figura 3.13: Fluxos de Feedforward e Feedback no SADI

Portanto, no *Framework SADI*, a lacuna entre o *Serviço de Aprendizagem* e o *Serviço de Tomada de Decisão* é reduzida para apenas uma contagem de *Tempo Contextual* T_C . Além disso, cada *Contexto* contém a semântica para avaliar se a *Percepção de Feedback Interior (PI)* significa uma *Percepção de Feedback Interior Bom/certo/sucesso (F_B)* ou uma *Percepção de Feedback Interior Ruim/errado/insucesso (F_R)*, dependendo dos resultados/efeitos esperados da *PA* realizada sobre ou relacionada com *PO*. Note que F_B e F_R são usados pelo *Serviço de Tomada de Decisão*.

Exemplo 2.2.9 Sobre a semântica da *Percepção de Feedback Interior (PI)*.

Considere o cenário do Exemplo 2.2.6, onde um andróide decide caminhar. No *Contexto* 'Movimento', dado a $PA = \text{'Mover.Caminhar.Passo'}$ relacionada com $PO = \text{'Limpo.Terreno'}$, o PI que mede o sucesso (isto é, F_B) pode ser um *Equilíbrio_Status* lido de um Sensor de Equilíbrio e um *Não_Estragado_Status* lido de um Sensor de Pé.

Observações:

O *Serviço de Aprendizagem* é projetado considerando que o *Framework SADI* opera num cenário de Aprendizagem por Reforço onde é possível aprender durante a execução. Adicionalmente, é assumido que o *Framework SADI* também possui a propriedade de **Presença**, isto é, a propriedade de estar dentro do *Ambiente*, como um participante/ator e simultaneamente como um observador (o que inclui a observação de si mesmo).

A organização de conhecimento em *Contextos* dentro de um *Espaço de Contexto* (Definição 2.2.2.3) fornece um modelo conceitual que orienta (durante a operação do *Framework SADI*) o crescimento da Representação de Conhecimento SADI (Definição 2.3). Devido às propriedades de *Presença* e de *Mudanças Incrementais* (Teorema 2.2.2), este crescimento adapta (de forma contínua e incremental) a Representação de Conhecimento inicial (do modelo conceitual) aos padrões de comportamento do *Ambiente* (complexo, dinâmico e incerto) onde se está inserido.

Além disso, a organização de conhecimento em *Contextos* dentro de um *Espaço de Contexto* é muito útil para controlar as relações semânticas e as relações de causa-e-efeito entre *PO*, *PA* e *PI*. Este controle também oferece suporte para a estratégia de incluir informações de contextos (numa perspectiva estatística, em Observação 2 da Definição 2.2.2.1).

Adicionalmente, o conceito de *Tempo Contextual* T_C reduz a lacuna entre o *Serviço de Tomada de Decisão* (Definição 2.2.3) e o *Serviço de Aprendizagem* para apenas uma contagem T_C , permitindo suas operações combinadas e incrementais em tempo real. De outra forma, esta redução da lacuna seria muito difícil se o processo de aprendizagem tivesse uma longa lacuna, isto é, se tivesse de fazer *off-line*: preparação de dados, pesquisa por padrões, avaliação e refinamento de conhecimento, etc. Uma longa lacuna faz perder um grande número de relacionamentos semânticos valiosos (dos contextos de origem dos dados) enquanto os dados estão sendo massivamente armazenados, antes da execução do processo de aprendizagem. Esta perda de semântica cria um dualismo (*trade-off*) entre as necessidades de: i) Coletar dados suficientes (para obter importância estatística); *versus* ii) Rastrear problemas não-estacionários, p.ex. usando técnicas de esmaecimento (*fading*).

3.2.3 O Framework SADI – Representação de Conhecimento

Definição 2.3 A Representação de Conhecimento SADI

O *Framework SADI* usa a *Arquitetura de Serviços SADI* (Definição 2.2) para construir e manter a **Representação de Conhecimento SADI**, a qual utiliza a sobreposição de *Símbolos Compostos* comuns (em Teorema 2.2.1) como base para representar três relacionamentos simultâneos:

1. O **Primeiro Relacionamento** é de dependência condicional entre *Símbolos Compostos* e seus componentes: os *Símbolos* e/ou *Símbolos Compostos* menores. Por exemplo, na Figura 3.7-B, pode ser representado $P(A|B) = P(A,B) / P(B)$ e $P(B|A) = P(A,B) / P(A)$. Em Relacionamentos Causais (Definição 2.3.1), é descrito como probabilidades e probabilidades condicionais são representadas;
2. O **Segundo Relacionamento** é sobre relacionamentos semânticos entre *Símbolos*. Apesar da organização hierárquica de *Símbolos* (e *Símbolos Compostos*), estes *Símbolos* podem ocorrer em múltiplos *Contextos*, e cada relacionamento entre um *Símbolo* componente e seu respectivo *Símbolo Composto* estabelece uma dimensão semântica. Portanto, a estrutura inteira que representa estes relacionamentos semânticos é de fato uma estrutura multidimensional. Isto é descrito nos Relacionamentos Semânticos Multidimensionais (Definição 2.3.2);
3. O **Terceiro Relacionamento** ocorre ao nível de estruturas de acesso aos dados, porque este relacionamento de representação de conhecimento é usado também como um índice de acesso às estruturas de dados que suportam o modelo conceitual. Esta indexação de acesso (implícita no modelo conceitual) é útil para carregar os dados sendo trabalhados, e isto é descrito em Estruturas de Acesso aos Dados (Definição 2.3.3).

Definição 2.3.1 Relacionamentos Causais

Dado que:

- Um *Contexto* (Definição 2.2.2.1) é ativado se somente se: 1^o) ele recebe o *Fluxo de Percepção* (Definição 2.2.1.6); e 2^o) suas *Percepções Incertas* são dinamicamente definidas pelo *Casamento de Sintaxe e Semântica* (Definição 2.2.2.5);
- O *Serviço de Aprendizagem* (Definição 2.2.4) atualiza (em T_C) o conhecimento dos *Contextos* alcançados pelo *Fluxo de Feedforward* dentro do *Espaço de Contexto* (Definição 2.2.2.3); e, também atualiza (em T_C) o conhecimento dos *Contextos Estendidos* (Definição 2.2.2.6) dentro da *Representação de Conhecimento SADI*; e,
- Esta atualização é feita através do controle semântico (fornecido pelo *Serviço de Detecção de Contexto*, Definição 2.2.2) e a relação de causa-e-efeito entre os *Símbolos* (PO, PA, PI) do *Fluxo de Feedforward* (PO_{T_C}, PA_{T_C}) e o *Fluxo de Feedback* ($(PO_{T_C}, PA_{T_C}).PI_{T_C+i}$).

Portanto:

Cada *Símbolo S* alcançado por um *Fluxo de Percepção* (*Fluxo de Feedforward* ou *Fluxo de Feedback*) incrementa seu *Contador de Frequência Relativa* ($S.Countador$) que é usado como uma medida de probabilidade.

Nos *Contextos* que recebem o *Fluxo de Feedforward* dentro do *Espaço de Contexto*, apenas os *Símbolos S* alcançados pelo *Fluxo de Feedforward* incrementam seus $S.Countadores$.

Todavia, nos *Contextos Estendidos* (dos *Contextos* ativados, isto é, os *Contextos* que recebem o *Fluxo de Feedforward* e o *Casamento de Sintaxe e Semântica*), todos os *Símbolos S* alcançados (tanto no *Fluxo de Feedforward* quanto no de *Feedback*) incrementam seus $S.Countadores$.

Com relação ao *Espaço de Contexto*, que é o ponto de entrada comum de onde todos os *Contextos Estendidos* são derivados, os *Símbolos S* obtêm suas probabilidades $P(S)$ simplesmente calculando:

$$P(S) = S.Countador / T_S.Countador$$

(onde $T_S.Countador$ é o contador de *Tempo de Sistema*, Definição 2.2.1.3).

Da mesma forma, para *Símbolos S* dos *Contextos Estendidos*:

$$P(S) = S.Countador / T_C.Countador$$

(onde $T_C.Countador$ é o contador de *Tempo Contextual*, Definição 2.2.2.2).

O $T_S.Countador$ deve ser suficientemente grande, mas ele vai transbordar (*overflow*) depois de um longo período. Portanto, antes de ocorrer o *overflow*, o *Framework SADI* deve monitorar o $T_S.Countador$ e escolher um bom momento (p.ex. de baixa carga no sistema) para realizar uma operação de deslocamento à direita (*shift-right*, uma divisão rápida por dois) no $T_S.Countador$ e em todo contador de *Símbolo S* do *Espaço de Contexto*. Depois da operação de *shift-right*, cada *Símbolo S* antigo e não-usado (isto é, cujo $S.Countador = 0$) possui seu $S.Countador$ restaurado para 1 se este S ainda está instanciado em algum *Contexto Estendido* (isto é, seu $S.Countador_Referência \geq 1$). Caso contrário, se este S é um *Símbolo Composto*, então ele é removido do *Espaço de Contexto* porque ele é uma combinação (de *Símbolos*) antiga e não-usada.

O tamanho de cada $T_C.Countador$ é pré-definido no *Framework SADI* mas ele pode ser redefinido pela aplicação baseada no SADI, porque este tamanho influencia a taxa de esmaecimento (*fading rate*) na qual *Símbolos* antigos e não-usados são

“esquecidos” nos *Contextos Estendidos* onde eles pertencem. Esta abordagem contextual minimiza a dificuldade de rastrear problemas não-estacionários, porque o T_C .Contador é apenas incrementado quando seu respectivo *Contexto Estendido* é ativado. Portanto, sempre antes do T_C .Contador transbordar (*overflow*), o *Framework SADI* realiza uma divisão *shift-right* neste T_C .Contador e em todo S .Contador de *Símbolo* deste *Contexto Estendido*. Depois da operação de *shift-right*, cada *Símbolo S antigo e não-usado* (isto é, cujo S .Contador = 0) é removido do *Contexto Estendido* e seu S .Contador_Referência é decrementado no *Espaço de Contexto* (adicionalmente, ele também é removido do *Espaço de Contexto* se seu S .Contador_Referência = 0). É importante notar que um dado *Símbolo S* do *Espaço de Contexto* possui S .Semântica e S .Contador diferentes para cada *Contexto* onde S ocorre (conforme descrito em Observação 1, de Relacionamentos Semânticos Multidimensionais, Definição 2.3.2).

Tanto no *Espaço de Contexto* quanto nos *Contextos Estendidos*, o *Fluxo de Percepção* é hierárquico, e para cada sub-árvore da hierarquia, os *Símbolos* componentes ($S_{a1}, S_{a2}, \dots, S_{an}$) e o *Símbolo Composto* ($S_q = \{S_{a1}, S_{a2}, \dots, S_{an}\}$) representam probabilidades e probabilidades condicionais (da dependência condicional entre S_a e S_q) – cujos S .Contador são atualizados/incrementados pelos *Fluxos de Percepção* (Figura 3.14-A). A **Regra da Cadeia** garante o crescimento de *Símbolos Compostos* de qualquer tamanho (Figura 3.14-B).

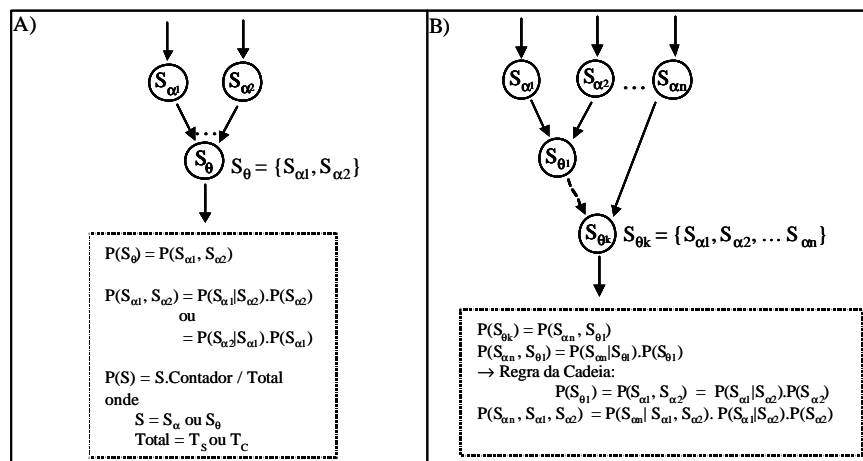


Figura 3.14: Probabilidades e Probabilidades Condicionais

Assim como no Sistema Psi (ψ) de Freud (em [MAC 2003]), a *Representação de Conhecimento SADI* não representa toda combinação teoricamente possível de *Símbolos*, porque isto levaria a uma representação com tamanho exponencial (mas extremamente esparsa em termos de uso). Portanto, a *Representação de Conhecimento SADI* representa apenas os *Símbolos* que estão conceitualmente definidos no *Espaço de Contexto* e suas combinações posteriores de *Símbolos Compostos* percebidos – ao invés de todas as combinações teoricamente possíveis de *Símbolos Compostos*. Note que, a *Representação de Conhecimento SADI* inicial (conceitualmente definida no *Espaço de Contexto*) é adaptada (de forma contínua e incremental) aos padrões de comportamento particulares do *Ambiente*, o qual assume-se que exibe a propriedade de Sistema Esparsos (em Definição 2.2.2).

É muito importante enfatizar que a *Representação de Conhecimento SADI* não usa o conceito de variáveis que podem variar seus estados (p.ex. $A = \text{‘Verdadeiro’}$, $A = \text{‘Falso’}$). Ao contrário, cada estado de variável é representado como um *Símbolo*, então ‘ A ’ significa apenas a ocorrência de $A = \text{‘Verdadeiro’}$. Da mesma forma, um *Símbolo*

Composto de qualquer tamanho $((S_1.S_2)...S_n)$ significa apenas a ocorrência de ' $S_1=Verdadeiro, S_2= Verdadeiro \dots S_n= Verdadeiro$ '. Se existe a necessidade de representar ' $A = Falso$ ' (ou qualquer variação de estados representados dentro de um *Símbolo Composto*) então *Símbolos* adicionais devem ser usados. Note que este uso de *Símbolos* permite a representação de dependência condicional não apenas entre variáveis do domínio, mas também a dependência condicional específica entre estados particulares das variáveis do domínio (o que é apropriado para *Ambientes* que também exibem a propriedade de *Sistemas Esparsos*, em Definição 2.2.2).

Portanto, a *Representação de Conhecimento SADI* fornece uma aproximação numérica de uma tabela de Distribuição de Probabilidade Conjunta (DPC). Uma Rede Bayesiana [PEA 88] possui uma representação exata de uma DPC, ao usar uma coleção de declarações de independências condicionais, onde as tabelas de Distribuição de Probabilidades Condicionais armazenam todas as permutações possíveis de estados dos nodos "pais" (os nodos que tem uma influência causal nos nodos "filhos") nesta rede.

Sempre que uma DPC sobre um domínio é construída, então ela pode ser usada para responder qualquer consulta sobre o domínio [RUS 95]. Uma vez que a *Representação de Conhecimento SADI* é uma aproximação numérica de uma DPC, sempre que uma aplicação baseada no SADI requisita (via consulta) uma probabilidade condicional que não está representada num dado *Contexto Estendido*, então o sistema de *Representação de Conhecimento SADI* fornece uma **Resposta Aproximada** – que é uma aproximação onde se faz a suposição ingênua (*naive*) de que alguns *Símbolos* são condicionalmente independentes, isto é, aqueles *Símbolos* dos subconjuntos não-representados (envolvidos na consulta). A Figura 3.15-A mostra o caso de uma probabilidade condicional representada $P(C/A,B)$ e a Figura 3.15-B mostra o caso de uma probabilidade condicional não-representada $P(A/B,C)$ – que é aproximada substituindo $P(B,C)$ por $P(B).P(C)$, supondo que B e C sejam condicionalmente independentes.

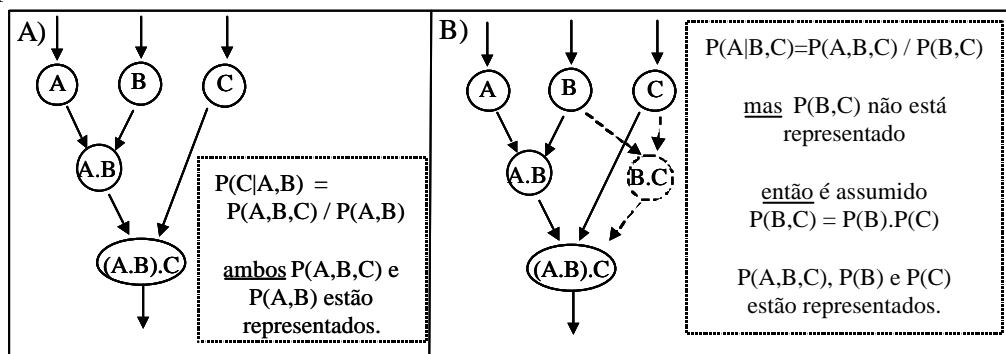


Figura 3.15: Probabilidades Condicionais Representadas e Não-representadas

Para fornecer esta *Resposta Aproximada*, se estes *Símbolos* alternativos B e C não são encontrados primeiramente no respectivo *Contexto Estendido* (o qual provê melhores valores de aproximação), então o sistema de *Representação de Conhecimento SADI* procura por $P(B,C)$ ou $P(B).P(C)$ no *Espaço de Contexto*. O *Espaço de Contexto* provê uma aproximação generalizada porque ele é o ponto de entrada comum de onde todo *Contexto Estendido* é derivado, então ele contém os *Símbolos* cujos S .Contadores são atualizados continuamente pelo *Fluxo de Percepção*. Note que, um *Contexto* (e então seu respectivo *Contexto Estendido*) é apenas ativado se ele recebe o *Fluxo de Percepção* e suas *Percepções Incertas* são definidas dinamicamente pelo *Casamento de Sintaxe e Semântica*.

Se os *Símbolos* alternativos não são encontrados em nenhum lugar, então a aplicação baseada no SADI recebe um Não_Encontrado_Status e então decide como lidar com esta exceção (p.ex. usando uma ação *default*, ou escolhendo uma ação candidata randomicamente, etc).

Definição 2.3.2 Relacionamentos Semânticos Multidimensionais

Uma vez que o conceito de Rede Semântica é incluído e estendido na Definição 2.2.2.1 de *Contexto*, então a semântica de um *Símbolo* depende de como este *Símbolo* está relacionado com outros *Símbolos*. No *Framework* SADI, os relacionamentos entre *Símbolos* podem ser via: 1^a) um *Símbolo Composto* (em Definição 2.2.1.2); 2^a) um *Relacionamento-Transição* (em Definição 2.2.2.4); e, 3^a) uma relação de causa-e-efeito entre $(PO_{Tc}.PA_{Tc})$ e $((PO_{Tc}.PA_{Tc}).PI_{Tc+1})$, em Definição 2.2.4.

Perspectiva de Símbolo Composto

Primeiramente os Relacionamentos Semânticos Multidimensionais são descritos na perspectiva de *Símbolo Composto*.

Considerando que: 1^a) *Símbolos* (e *Símbolos Compostos*) podem ocorrer em múltiplos *Contextos*; e, 2^a) cada relacionamento entre um *Símbolo* componente e seu respectivo *Símbolo Composto* define uma dimensão semântica; então a estrutura inteira resultante é multidimensional. Todavia, em T_S , apenas uma propagação hierárquica (do *Fluxo de Percepção*, em Teorema 2.2.1) ocorre dentro desta estrutura multidimensional, onde os *Símbolos* percebidos são distribuídos para cada um de seus relacionamentos semânticos.

Na Figura 3.16-A, a propagação hierárquica (em T_S) é ilustrada pelas setas pretas mais largas (e as setas pretas mais finas ilustram a distribuição dos *Símbolos* para outros *Contextos*), enquanto que as setas cinzas ilustram outros caminhos existentes (de propagação hierárquica) que não são ativados (em T_S).

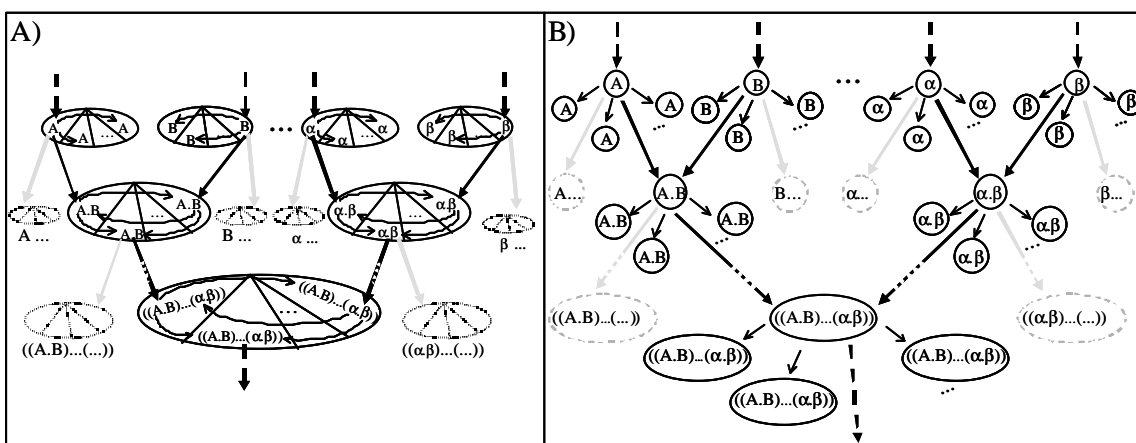


Figura 3.16: Visão Multidimensional da *Representação de Conhecimento SADI*

A Figura 3.16-B mostra uma abstração da estrutura da Figura 3.16-A, sem exibir a organização de Transmissões Vertical e Horizontal (em Definição 2.1.3).

Na Figura 3.16-B, pode ser observado que o relacionamento multidimensional entre *Símbolos* (e *Símbolos Compostos*) é similar a um Quadro (*Frame*, da Inteligência

Artificial), considerando a característica de que um atributo de um Quadro pode ser outro Quadro, p.ex. $((A.B)...(\alpha.\beta))$ é um atributo de $(A.B)$, que é um atributo de (A) .

Ainda considerando o relacionamento multidimensional mostrado na Figura 3.16-B, ele também é similar ao Modelo de Dados *Snowflake* (de *Data Warehouse*), considerando a característica de que uma Tabela Dimensão (isto é, uma das “pontas da estrela”) de uma Tabela Fato (isto é, o “núcleo da estrela”) pode ser outra Tabela Fato, p.ex. $(A.B)$ é uma Tabela Dimensão de (A) e $(A.B)$ é também uma Tabela Fato que tem $((A.B)...(\alpha.\beta))$ como uma Tabela Dimensão.

Os Quadros (*Frames*) podem ser compreendidos pela abordagem da Teoria dos Conjuntos [RIC 93, p.297]. Aqui, também é considerado que a outra similaridade entre a *Representação de Conhecimento SADI*, Redes Semânticas, Quadros e Modelo de Dados *Snowflake* é: suas raízes na Teoria dos Conjuntos.

Perspectiva de Relacionamento-Transição

Agora os Relacionamentos Semânticos Multidimensionais são descritos na perspectiva de *Relacionamento-Transição*.

Dado um *Símbolo* π no *Espaço de Percepção*, ele participa num conjunto de *Símbolos Compostos* $\{(\pi.\alpha),(\pi.\beta), \dots(\pi.\gamma)\}$, onde cada elemento representa uma dimensão semântica de π (ver Figura 3.17). Portanto, em T_S , a percepção da sintaxe do *Símbolo* π é casada com a semântica do *Símbolo* percebido α , através do *Casamento de Sintaxe e Semântica* iniciado (via o *Relacionamento-Transição* $\alpha:?\text{--}\alpha$) no *Estado de Ação* A_k ativo do *Modelo de Mundo*. Depois, em $T_{S+\Delta}$ ($\Delta = 1,2,\dots,n$), uma seqüência de transições entre *Estados de Ação* muda o *Estado de Ação* ativo de A_k para A_j . Portanto, a percepção da sintaxe do *Símbolo* π é agora (em $T_{S+\Delta}$) casada com a semântica do *Símbolo* percebido γ , através do *Casamento de Sintaxe e Semântica* iniciado (via o *Relacionamento-Transição* $\gamma:?\text{--}\gamma$) no atual *Estado de Ação* ativo A_j .

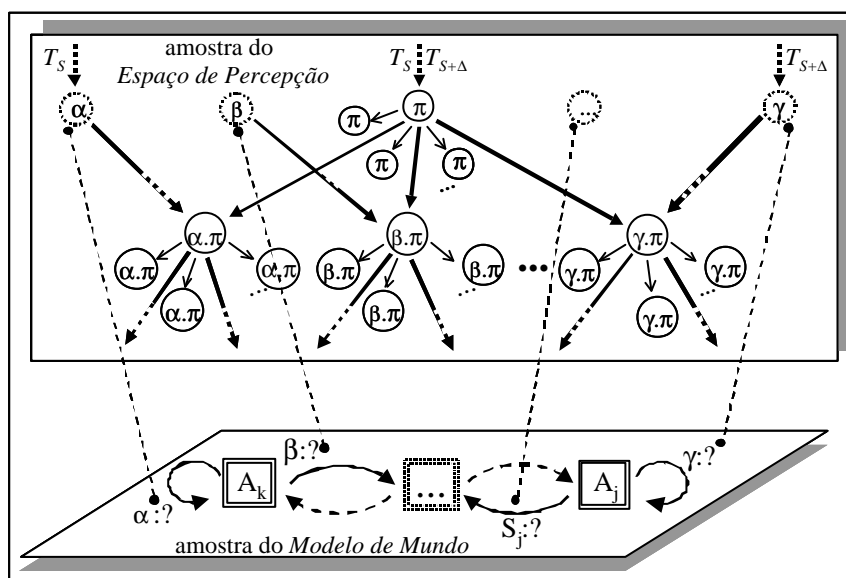


Figura 3.17: Visão Multidimensional sob o *Relacionamento-Transição*

Exemplo 2.2.10 Sobre Relacionamentos Semânticos Multidimensionais.

Considere novamente o cenário do Exemplo 2.2.6. Em T_S , o objetivo de caminhar do andróide casa a sintaxe da percepção de ‘Terreno’ com a semântica do

Relacionamento-Transição 'Limpo:?'–'Limpo'. Depois, em $T_{S+\Delta}$ ($\Delta = 1, 2, \dots, n$), suponha que o andróide parou de caminhar mas ele ainda está percebendo a sintaxe do *Símbolo* 'Terreno'. Todavia, agora o objetivo do andróide é analisar a resistência do terreno. Portanto, o atual *Estado de Ação* ativo provê (via o *Relacionamento-Transição* 'Resistente:?'–'Resistente') a semântica a ser casada com 'Terreno', se o *Símbolo* 'Resistente' é também percebido em $T_{S+\Delta}$.

Perspectiva de Causa-e-Efeito

Finalmente, os Relacionamentos Semânticos Multidimensionais são descritos na perspectiva das relações de causa-e-efeito entre o $(PO_{T_c}.PA_{T_c})$ e o $((PO_{T_c}.PA_{T_c}).PI_{T_{c+1}})$, em Definição 2.2.4.

Dado que os conceitos de PO , PA , e PI (em Definição 2.2.1.5, e Observação 2 da Definição 2.2.2.6) fornecem a base (juntamente com a Definição 2.2.2.2 de *Tempo Contextual* T_c) que provê suporte para as operações incrementais do *Framework* SADI (ao reduzir a lacuna entre o *Serviço de Tomada de Decisão* e o *Serviço de Aprendizagem* para apenas uma contagem T_c), portanto o relacionamento entre PO , PA e PI compõe o cubo básico conceitual da Representação de Conhecimento Multidimensional. Note que, cada símbolo de ação pode estar relacionado com diferentes objetos, e também cada objeto pode estar relacionado com diferentes ações. A Figura 3.18 ilustra um símbolo de uma certa ação A_a que foi aplicada (em diferentes momentos) a um conjunto de objetos (O_1, O_2, \dots, O_n) .

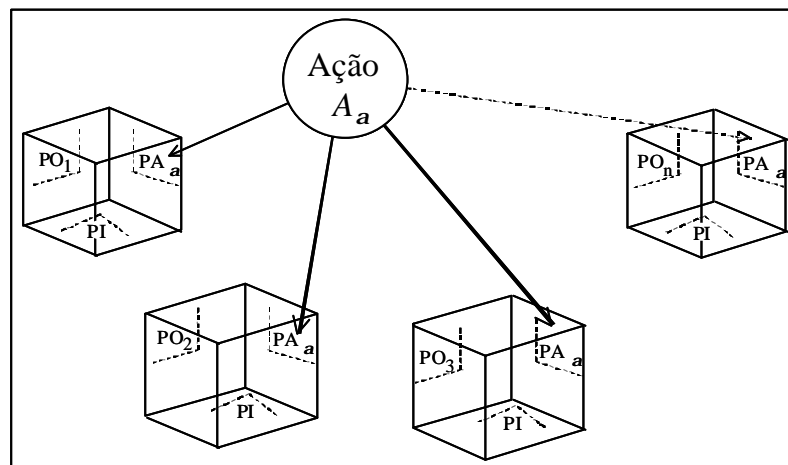


Figura 3.18: Visão Multidimensional sob a Perspectiva de Causa-e-Efeito

Observação 1)

Um dado *Símbolo* S do *Espaço de Contexto* possui diferentes S .Semântica e S .Contador $_{T_s}$ para cada *Contexto* onde S ocorre. Quando um *Símbolo Composto* S_C é formado no *Espaço de Contexto*, ele também possui diferentes S_C .Semântica e S_C .Contador $_{T_s}$ para cada *Contexto* onde S_C ocorre (note que o Contador $_{T_s}$ é baseado no *Tempo de Sistema* T_s). Uma vez que um *Contexto* é ativado, seu respectivo *Contexto Estendido* instanciado contém *Símbolos* S e *Símbolos Compostos* S_C cujas S .Semântica e S_C .Semântica são as mesmas do *Contexto* de origem, todavia seus S .Contador $_{T_c}$ e S_C .Contador $_{T_c}$ são baseados no *Tempo Contextual* T_c .

Definição 2.3.3 Estruturas de Acesso aos Dados

O *Terceiro Relacionamento* da *Representação de Conhecimento SADI* ocorre ao nível de Estruturas de Acesso aos Dados. O modelo lógico da *Representação de Conhecimento SADI* é diretamente mapeado no modelo físico, porque:

- Um único relacionamento entre dois *Símbolos* no modelo lógico representa simultaneamente um Relacionamento Causal (Definição 2.3.1) e uma dimensão dos Relacionamentos Semânticos Multidimensionais (Definição 2.3.2); e,
- Este único relacionamento é diretamente representado por um índice de acesso à estrutura de dados que suporta o modelo lógico.

Portanto, esta indexação de acesso (implícita no modelo lógico) é útil para carregar os dados sendo trabalhados (p.ex. via *look-ahead buffers*), especialmente se for considerada a propriedade de *Mudanças Incrementais* no *Espaço de Contexto* (em Definição 2.2.2.3, em Teorema 2.2.2).

Exemplo 2.2.11 Sobre Estruturas de Acesso aos Dados.

Considere o exemplo (baseado em [CAS 97,Capítulo 8]) de uma Rede Bayesiana mostrada na Figura 3.19-A. Para ser usada eficientemente, esta Rede Bayesiana é transformado numa Árvore de Junção composta por clusters chamados Cliques (Figura 3.19-B).

Dada a Rede Bayesiana da Figura 3.19-A, a fatorização de sua Distribuição de Probabilidade Conjunta (DPC) resulta em:

$$P(A,B,C,D,E,F) = P(A).P(B|A).P(C|A).P(D|B).P(E|B,C).P(F|C)$$

Uma vez que B e C são condicionalmente independentes, $P(B|A)$ pode ser reescrito como $P(B|C,A)$ e se $P(A)$ for movido para a terceira posição, então é obtido $P(B|C,A).P(C|A).P(A)$, que pode ser reescrito como $P(B,C,A)$:

$$P(A,B,C,D,E,F) = P(B,C,A).P(D|B).P(E|B,C).P(F|C)$$

Finalmente, se for usado $P(\mathbf{a}|\mathbf{b}) = P(\mathbf{a},\mathbf{b}) / P(\mathbf{b})$, o resultado é:

$$P(A,B,C,D,E,F) = P(B,C,A) \frac{P(D,B).P(E,B,C).P(F,C)}{P(B) P(B,C) P(C)}$$

As relações de conjunto da fatorização desta DPC é mostrada na Figura 3.19-C, que pode ser considerada como uma projeção destas relações de conjunto num plano. Na *Representação de Conhecimento SADI*, as relações de conjunto são expandidas no espaço multidimensional (conforme ilustrado na Figura 3.19-D), onde os *Fluxos de Percepção* mantêm as probabilidades atualizadas usando serviços incrementais e contínuos. Por clareza, as relações de conjunto da Figura 3.19-D não mostram todos os *Símbolos* (e *Símbolos Compostos*) necessários para expressar toda a variação de estados das variáveis randômicas da Rede Bayesiana correspondente.

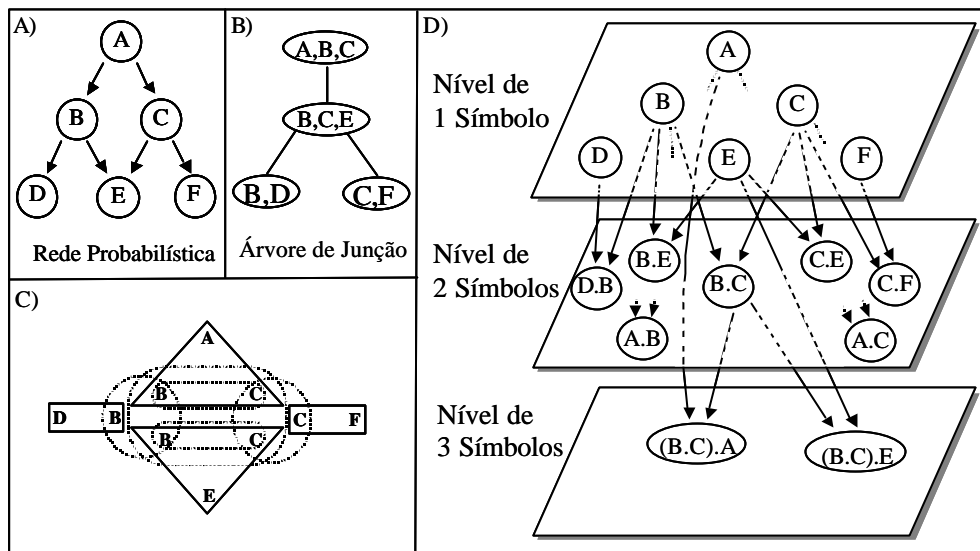


Figura 3.19: A) Exemplo de Rede Bayesiana; B) Exemplo de Árvore de Junção que resultou desta Rede Bayesiana; C) Visualização das relações de conjunto resultadas da fatorização da DPC. D) Relações de conjunto na *Representação de Conhecimento SADI*

Comentários sobre a Representação de Conhecimento

Soluções para o problema de representar conhecimento para raciocínio mediante incerteza tem sido desenvolvido desde o início da Teoria da Probabilidade. Teorias formais foram introduzidas depois, tal como Teoria da Evidência [Dempster, 1968; Shafer, 1976 *Apud* FUN 86], Teoria da Possibilidade [ZAD 78] e Fatores de Certeza [DAV 77]. Além destas abordagens numéricas, lógicas não-monotônica foram adaptadas para superar as limitações da lógica de primeira ordem para poder lidar com incompletude (*incompleteness*) e incerteza, tal como Lógica *Default* [REI 80], Argumentação [94; *Apud* PAR 98], e Sistemas de Manutenção de Verdade [DOY 79, KLE 86]. Mais recentemente, a habilidade humana para raciocinar mediante informação incerta inspirou [PEA 88] a desenvolver uma teoria baseada em relacionamentos de dependência condicional. Esta teoria provê suporte para uma estrutura chamada Rede Bayesiana (também chamada de Rede de Crença), que tem atraído o interesse de pesquisadores e da indústria de software.

Todavia, as raízes destas abordagens anteriores pertencem a um paradigma no qual existe uma longa lacuna entre o processo de aprendizagem e o processo de decisão/execução. Esta longa lacuna geralmente inclui: preparação de dados, pesquisa por padrões, avaliação e refinamento de conhecimento, etc. No caso de Redes Bayesianas, o processo de aprendizagem básico é uma instância de um problema *NP-Hard* [CHI 94].

Em [MAC 2003] também é descrito que a experimentação e assimilação de todos os padrões de percepção dentro de um ambiente onde um indivíduo está inserido, não representa uma possibilidade real concreta, considerando a economia de neurônios usada pelo sistema Psi (ψ). Portanto, a organização do conhecimento em *Contextos* é muito importante para criar representações localmente estruturadas do *Ambiente* do mundo real (exterior ao *Corpo*) dentro de um *Modelo de Mundo* menor e compacto (dentro do *Corpo*), porque os *Contextos* delimitam o escopo dos relacionamentos. Numa

perspectiva física, as dimensões de tempo e espaço e a lei de causa-e-efeito delimitam o escopo dos *Contextos* de relacionamentos entre eventos. Para os humanos, o escopo dos *Contextos* é também delimitado numa perspectiva comportamental – de acordo com [BER 70] – as crianças tem suas mentes programadas pelos seus pais (ou por aqueles que as educaram), e este programa mental (geralmente) estabelece um padrão de comportamento para o resto da vida. Mesmo com a complexidade, o dinamismo e a incerteza da vida, é possível observar que todas as ações de um indivíduo são influenciadas pelos “atratores” ou “repulsores” definidos no programa mental. Usando conceitos de ciência da computação, este programa mental possui uma similaridade funcional com *Frameworks* de Objetos do Paradigma de Orientação à Objetos, e também com Roteiros (*Scripts*) da Inteligência Artificial [RIC 93,pp.326].

Finalmente, a abordagem do *Framework* SADI de decisões incrementais e aprendizagem incremental durante a execução, combinada com a sinergia provida pelos três relacionamentos simultâneos (de 1^o Relacionamentos Causais; 2^o Relacionamentos Semânticos Multidimensionais; e 3^o Estruturas de Acesso aos Dados), torna o *Framework* SADI conceitualmente similar a (o que poderia se chamar de) um *Datawarehouse* Orientado a Objetos que opera de forma contínua e incremental usando o *Framework* SAIC (assim como [ABO 99], que constrói histogramas auto-ajustáveis para serem usados em otimização de consultas à banco de dados). Esta similaridade conceitual se refere a: 1) Dados orientados a contexto; 2) Muitos dados sumarizados e sintetizados; 3) Muitas estruturas de índices; 4) Representação de conhecimento multidimensional; e, 5) Armazenamento relativamente grande – proporcional ao escopo do domínio. Nós também acreditamos que esta similaridade conceitual também pode ser aplicada ao cérebro humano – que é obviamente mais complexo e poderoso.

3.2.4 Computação Contextual e Metodologia Contextual

Definição 2.4.1 Computação Contextual

A **Computação Contextual** usa o *Framework* SADI, isto é, a *Computação Contextual* usa a *Arquitetura de Serviços SADI* (Definição 2.2) para construir e manter a *Representação de Conhecimento SADI* (Definição 2.3).

Definição 2.4.2 Metodologia Contextual

A **Metodologia Contextual** consiste em duas fases principais que são realizadas continuamente como um ciclo evolutivo:

1. **Fase de Definição**, que é o projeto e implementação de um Sistema de Aplicação baseado no *Framework* SADI;
2. **Fase de Operação**, onde um Sistema de Aplicação baseado no *Framework* SADI complementa (isto é, estende e adapta) as definições iniciais (da *Fase de Definição*) através da interação com o *Ambiente*.

4 ESTUDO DE CASO: AGENTE DE SOFTWARE

Com relação à representação de conhecimento incerto e à tomada de decisão mediante incerteza, o Capítulo 2 revisou alguns dos principais fundamentos de Redes Probabilísticas (ou Bayesianas) e analisou a dificuldade que elas apresentam para controlar uma semântica de hierarquia de classes (gerando o Problema da Lacuna, descrito na Seção 2.8). Portanto, para minimizar o Problema da Lacuna, o Capítulo 3 descreveu o *Framework* SADI juntamente com a prova teórica de que ele preserva as propriedades do *Framework* SAIC – o qual serve de base para o SADI.

O Agente de Software deste estudo de caso visa ilustrar alguns dos principais conceitos definidos no *Framework* SADI, os quais são instanciados e adaptados para um cenário de consumo de informação que inclui:

- Incerteza: segundo [HOR 97], a identificação de preferências e intenções de usuários (durante a interação destes com sistemas de comunicação), usando evidências como ações do usuário e os contextos destas ações, é um problema típico de tomada de decisão mediante incerteza;
- Complexidade, Dinamismo e Distribuição: conforme descrito nas seções seguintes, o cenário deste Agente de Software é uma instância da definição de *Ambiente* (Definição 1.3, Capítulo 3) do *Framework* SADI, porque este cenário é de um *Ambiente* Virtual composto de diversas funcionalidades que estão distribuídas nas aplicações e ferramentas de software deste cenário. Por exemplo, a funcionalidade ‘Pesquisar’ existe no editor de texto, correio eletrônico, navegador *web*, gerenciador de arquivos e na agenda (*scheduler*). Ou seja, esta funcionalidade está distribuída no *Ambiente* Virtual, mas ela reflete de alguma forma o interesse do usuário, o qual é muito dinâmico. Avaliar a intensidade e a duração deste interesse depende dos contextos das ações e isto gera uma complexidade muito grande - principalmente na coleta, organização e interpretação dos eventos sendo monitorados.

Antes de iniciar a modelagem conceitual do estudo de caso, na Seção 4.1 vamos descrever o objetivo deste agente de software. Depois, a Seção 4.2 apresenta uma análise sobre o conteúdo (de informações que o agente vai perceber do *Ambiente*) e sua relação com a representação de conhecimento. Em seguida, a *Arquitetura de Serviços* SADI (Definição 2.2, Cap.3) é instanciada no cenário desta aplicação. Portanto, a descrição do Agente de Software é feita na seguinte ordem: Na Seção 4.3, o *Serviço de Percepção do Ambiente* (Definição 2.2.1, Cap.3); Na Seção 4.4, o *Serviço de Detecção de Contexto* (Definição 2.2.2, Cap.3); Na Seção 4.5, o *Serviço de Aprendizagem* (Definição 2.2.4, Cap.3); e, na Seção 4.6, o *Serviço de Tomada de Decisão* (Definição 2.2.3, Cap.3).

4.1 O Objetivo do Agente de Software

A motivação deste trabalho, apresentada no Capítulo 1, menciona a importância do consumo de informação baseado na Internet e destaca também a dificuldade que existe para as pessoas encontrarem ou selecionarem as informações desejadas. Procurando diminuir esta dificuldade, ainda no Capítulo 1, foram citados três trabalhos desenvolvidos pelo nosso grupo de pesquisa: o primeiro trabalho propõe Canais de Informação com Serviços de Filtragem no software Direto [BAL 2002] (feito num projeto conjunto com o Governo do Estado do Rio Grande do Sul); e, o segundo trabalho complementa o primeiro, ao descrever um agente de software do tipo assistente pessoal [MEL 2002], que monitora o comportamento do usuário do Canal de Informação e o auxilia na tarefa de atualizar seu perfil de consumo (isto é, seu foco de interesse). Já o terceiro trabalho [FRI 2002] complementa os dois anteriores, ao fazer *Datamining* nas bases de dados dos Canais de Informação para prover novas informações aos agentes assistentes dos usuários dos Canais. Por exemplo, é possível informar ao usuário de um certo Canal C_n (com um Filtro F_h sobre um assunto A_z), que $X\%$ dos outros usuários que filtram o mesmo assunto A_z no Canal C_n , também filtram um outro assunto A_y em outro Canal C_k . Ainda no escopo das nossas atividades de pesquisa, foi desenvolvido um Agente do Tipo Assistente de Anotação [BOR 2000], cuja funcionalidade está incluída na ambiente de software deste estudo de caso.

O objetivo do agente de software deste estudo de caso é similar ao do agente de software do tipo assistente pessoal desenvolvido em [MEL 2002] – o qual monitora o comportamento de um usuário de Canais de Informação com Serviços de Filtragem [BAL 2002], e auxilia este usuário na tarefa de atualizar seu perfil de consumo (isto é, as regras de filtragem que refletem o foco de interesse dentro de cada canal). Ou seja, a funcionalidade deste agente é de um **Assistente de Feedback**.

No cenário desta aplicação, o usuário pode se cadastrar em um ou mais Canais de Informação, e opcionalmente configurar um ou mais filtros para cada Canal cadastrado (visando aprimorar o foco do seu interesse). Nos Canais de Informação com Serviços de Filtragem, o usuário ainda possui a opção de fazer o *download* do Assistente de *Feedback*, que ficará instalado no computador do usuário para poder monitorar o comportamento deste.

Uma questão muito importante é considerada neste projeto. Trata-se da proteção da privacidade do usuário e também do controle da segurança no Serviço de Canais. Primeiramente, todo o código dos programas do Serviço de Canais assim como o código do Assistente de *Feedback* deve ser Software Livre (pelas vantagens apresentadas em [BOR 2003a]). Além disso, o Assistente de *Feedback* não coleta nenhuma informação pessoal do usuário (nome, endereço, etc), apenas coleta informação técnica que descreve a preferência do usuário com relação ao conteúdo que deseja receber. Mais ainda, para o Serviço de Canais, o usuário é apenas um número (que fica associado com as informações técnicas). Este número que identifica o usuário é fornecido pelo Serviço Associador (descrito em [BOR 2003b]), o qual associa este número com a identidade do usuário. O **Serviço Associador** é o único que pode revelar a identidade do usuário, caso exista uma determinação judicial para tal. Portanto, as informações técnicas sobre o comportamento do usuário podem ser usadas para o controle da segurança, mas a identidade do usuário só pode ser revelada para operações legalizadas de serviços de investigação legalmente instituídos.

Com relação ao monitoramento do comportamento do usuário, o Assistente de *Feedback* desenvolvido em [MEL 2002] monitora a interação do usuário com a ferramenta de correio eletrônico, a qual é a ferramenta cliente do Serviço de Canais. Já o Assistente de *Feedback* deste estudo de caso monitora o comportamento do usuário em várias outras ferramentas/aplicações, além do correio eletrônico (isto é detalhado nas seções seguintes). Ambos os Assistentes de *Feedback* fazem as mesmas ações, podendo sugerir ao usuário: i) a exclusão de **Termos** (isto é, palavras-chave) de um filtro; e/ou, ii) a inclusão de novos Termos num filtro. Considerando que os serviços SADI utilizam uma forte organização contextual, as ações de sugestão do Assistente de *Feedback* ocorrem em dois contextos de sugestão (ilustrados na Figura 4.1):

- **Contexto_de_Exclusão:** O Assistente de *Feedback* neste contexto faz o rastreamento dos Termos provavelmente inúteis (que já estão no filtro), e que estão causando a filtragem de notícias provavelmente inúteis. Portanto, o Assistente de *Feedback* monitora o comportamento do usuário com relação ao **Conteúdo Filtrado**, isto é, o conteúdo contido nas notícias recebidas do serviço de filtragem, para poder inferir quais são os Termos mais (provavelmente) inúteis, e assim sugerir a remoção destes Termos.
- **Contexto_de_Inclusão.** Aqui a situação é a inversa, o Assistente de *Feedback* neste contexto faz o rastreamento dos Termos provavelmente úteis (e que ainda não estão no filtro), e que poderiam causar a filtragem de notícias provavelmente úteis. Logo, o Assistente de *Feedback* monitora o comportamento do usuário com relação ao **Conteúdo não-Filtrado**, isto é, o conteúdo que ele está trabalhando/usando – exceto que este conteúdo não é proveniente do serviço de filtragem (pode ser o conteúdo de outras mensagens de *e-mail*, dos arquivos que edita, das páginas *web* que acessa, etc).

O uso destes dois contextos serve para minimizar a incerteza intrínseca neste processo de inferir prováveis preferências (e não-preferências) de usuário. Embora o Assistente de *Feedback* tenha certeza dos eventos que ele está monitorando, estes eventos servem apenas como indicadores e não como garantias. O uso dos contextos acrescenta uma maior semântica, pois a inferência que um Termo *T* não é útil (dado o comportamento do usuário com relação ao Conteúdo não-Filtrado) não implica que o Termo *T* seja necessariamente útil quando o usuário interage com Conteúdo Filtrado. Da mesma forma, a inferência que um Termo *T* não é inútil (dado o comportamento do usuário com relação ao Conteúdo Filtrado) não implica que o Termo *T* seja necessariamente inútil quando o usuário interage com Conteúdo não-Filtrado.

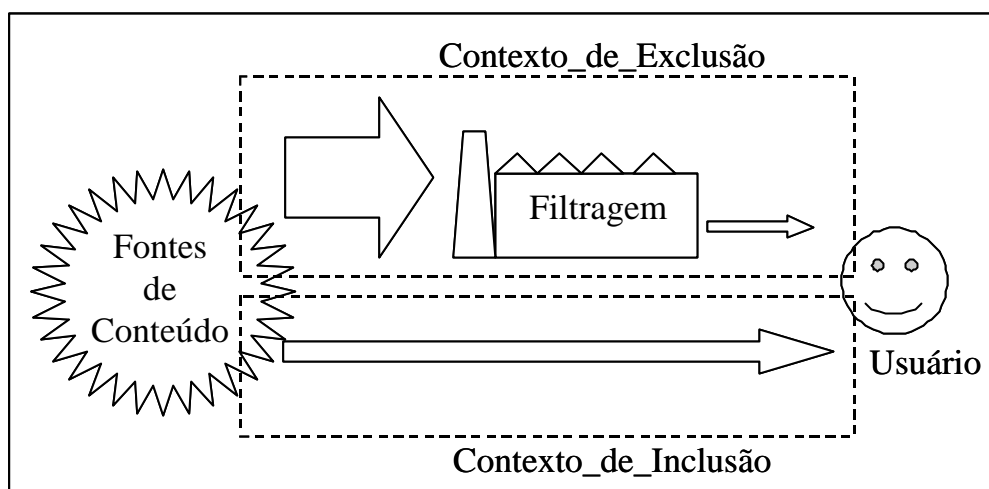


Figura 4.1: Contextos de Exclusão e Inclusão

Sobre o Conteúdo Filtrado, para acrescentar mais semântica no conteúdo e facilitar o seu controle de qualidade, ele está formatado obedecendo a padrões de Metadados, como os descritos na Seção 4.3. A Figura 4.2 mostra um exemplo de Conteúdo Filtrado que está formatado no padrão *Channel Definition Format* [CDF 97].

```

<!DOCTYPE Channel SYSTEM "http://www.w3c.org/Channel.dtd"/>
<Channel HREF="http://www.sports.com/sports.cdf" IsClonable=YES>
  <IntroUrl VALUE="http://www.foosports.com/channel-setup.html">
  <LastMod VALUE="1994.11.05T08:15-0500" />
  <Title VALUE="Sports" />
  <Abstract VALUE="The latest in sports" />
  <Author VALUE="Sports" />
</IntroUrl>
<Schedule>
  <EndDate VALUE="1994.11.05T08:15-0500" />
  <IntervalTime DAY=1 />
  <EarliestTime HOUR=12 />
  <LatestTime HOUR=18 />
</Schedule>
<Logo HREF="http://www.sports.com/logo.gif" Type="REGULAR" />
<Item HREF="http://www.foosports.com/articles/a1.html">
  <LastMod VALUE="1994.11.05T08:15-0500" />
  <Title VALUE="How to get the most out of your mountain bike" />
  <Abstract VALUE="20 tips on how to work your mountain-bike to
    the bone and come out on top." />
  <Author VALUE="FooSports" />
  <Usage VALUE="DesktopComponent" />
</Item>
</Channel>

```

Figura 4.2: Padrão *Channel Definition Format* para Conteúdo Filtrado

Além de sugerir a exclusão e/ou inclusão de Termos nos Filtros que o usuário criou para os Canais que cadastrou, o Assistente de *Feedback* deste estudo de caso também sugere a inclusão de Termos em outras atividades do usuário, por exemplo: anotando textos, enviando consultas para máquinas de busca (*search engine*) na *Web*, etc (maiores detalhes são apresentados na Seção 4.4).

Esta ação de sugerir Termos em outras atividades do usuário visa obter um *Feedback* de curto prazo, para poder subsidiar (refinando a sugestão sobre) a exclusão/inclusão de Termos nos Filtros, que é uma ação cujo *Feedback* é obtido num prazo maior. Normalmente, a ação do usuário criar um Filtro *F* num Canal *C* reflete um interesse mais duradouro, ao contrário da ação de consultar uma máquina de busca – que não garante que o assunto consultado reflete um interesse mais duradouro).

Todavia, para não aborrecer o usuário, este *Feedback* de curto prazo é obtido numa janela de ***Feedback Rápido***, ou seja, é uma janela cuja ação *default* é cancelar caso o usuário não clique nela (ou seja, o usuário não precisa interromper o que estava fazendo antes). Somente se o usuário achar a sugestão interessante, aí sim ele clica na janela de *Feedback Rápido*.

Portanto, esta Seção apresentou o objetivo deste Assistente de *Feedback*. Nas Seções seguintes, maiores detalhes são gradualmente descritos.

4.2 Metadados: Representação e Intercâmbio

Esta seção descreve o tipo de Conteúdo a ser percebido pelos sensores do Agente de Software (a serem apresentados no Serviço de Percepção do Ambiente, Seção 4.3) e comentar como a organização dos objetos de informação (isto é, os conteúdos) em hierarquias de classes facilita o processo de representação de conhecimento contextual do Agente de Software.

Durante as atividades de pesquisa, um dos nossos primeiros trabalhos [BOR 99] analisou como é feita a representação de conteúdo (de informações na *Web*) e qual é a sua relação com a representação de conhecimento. Desde então, pela força da Internet e da *Web*, tornaram-se comuns (nos ambientes de software) os padrões que representam conteúdos (de objetos de informação) e que também permitem a manipulação e o intercâmbio destes conteúdos.

Quanto à estruturação dos documentos, a linguagem *XML* (*Extensible Markup Language*) [BRA 98] forma a base sintática que sustenta todas as estruturações e formatos (existentes e a serem criados) por aplicações na *Web*. O modelo *RDF* (*Resource Description Framework*) [LAS 99] se apoia na *XML* e acrescenta um *framework* conceitual e abstrato para definir semântica nos dados *XML*. Dentre estes dados, estão os metadados [BOR 99,cap.3], os quais permitem uma variedade de aplicações, como as que são mostradas no topo da Figura 4.3.

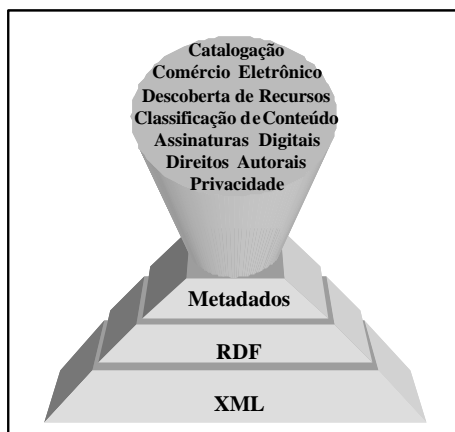


Figura 4.3: A Base dos Metadados e Suas Aplicações

Fonte: [BOR_99,p.33]

O Assistente de *Feedback* deste estudo de caso está relacionado com a aplicação ‘Descoberta de Recursos’ (na Figura 4.3), uma vez que ajuda o usuário a manter atualizado seu foco de consumo de informação, monitorando o comportamento do usuário com relação ao conteúdo que consome (na forma de metadados).

Embora não esteja no escopo do projeto do Assistente de *Feedback* a definição de uma ontologia, é preciso considerar a necessidade de seguir uma, pois a definição eficaz da semântica dos metadados está fortemente associada com o uso de ontologias. Em [DUV 99] é afirmado: “A força da estrutura de metadados está no fato do criador e do pesquisador usarem os mesmos termos”. O uso de uma mesma ontologia, tanto para quem disponibiliza um termo, quanto para quem o consulta, é vital para que exista uma convergência entre ambos. Ainda na mesma referência é afirmado que “o desenvolvimento de metadados será por seleção natural, onde cada comunidade desenvolverá o seu conjunto” (...) e “é necessário um nível básico de estabilidade para garantir a similaridade de termos entre as comunidades.” Na interseção de todos os conjuntos de metadados deverá existir um subconjunto que garanta a integração semântica dos metadados das diferentes comunidades – conforme mostra a Figura 4.4.

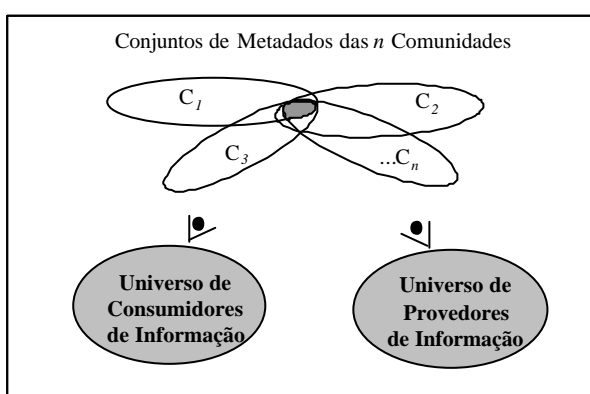


Figura 4.4: Visão dos Conjuntos de Metadados

Fonte: [BOR 99,p.34]

Além de auxiliar na definição da semântica dos metadados, uma ontologia está fortemente relacionada com a representação de conhecimento. Ao escrever sobre a Causalidade, [PEA 2000,p.25] expõe que: “Relacionamentos causais são ontológicos, descrevendo restrições físicas e objetivas do nosso mundo, enquanto que relacionamentos probabilísticos são epistemológicos, refletindo o que nós sabemos ou

acreditamos sobre o mundo.” Uma vez que a Representação de Conhecimento do *Framework SADI* (usada pelo Assistente de *Feedback*) representa tanto as relações causais quanto as relações de hierarquias de classes, fica mais nítida ainda a necessidade de usar ontologias.

De acordo com [SWA 99], “uma ontologia provê a estrutura básica ou armadura em torno da qual uma base de conhecimento pode ser construída. Uma ontologia provê um conjunto de conceitos e termos para descrever algum domínio, enquanto uma base de conhecimento usa estes termos para representar o que é verdadeiro sobre algum mundo real ou hipotético.” Pode-se concluir que o desenvolvimento de ontologias facilita:

1. A modelagem de bases de conhecimentos;
2. O desenvolvimento de metadados na *Web*;
3. A extração/geração de conhecimento de metadados *Web* por agentes (e outras aplicações de IA);
4. A representação e o intercâmbio de conhecimento entre agentes (e outras aplicações de IA).

Estes itens facilitados estão representados respectivamente pela numeração indicada na Figura 4.5, onde é ilustrada a influência das ontologias neste contexto de metadados e representação de conhecimento.

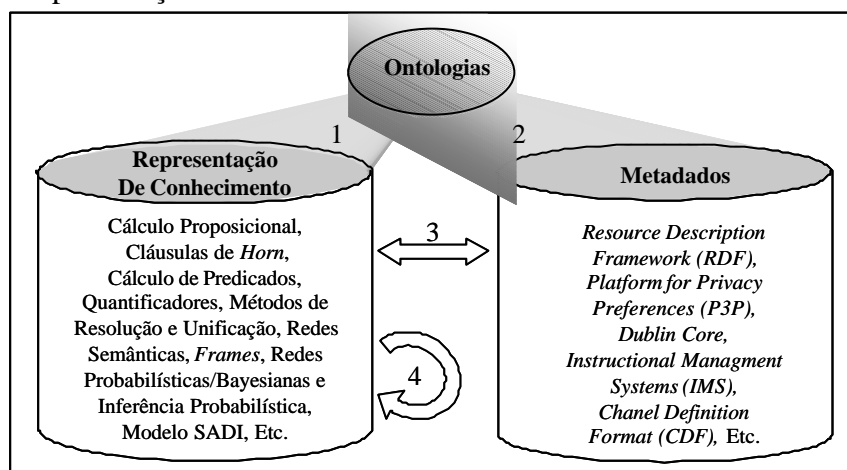


Figura 4.5: Influência das Ontologias

Fonte: [BOR 99,p.35]

Na Seção seguinte (4.3), é descrito como o Assistente de *Feedback* monitora o comportamento do usuário, capturando eventos e extraindo conteúdo. Quanto à extração de conteúdo, é apresentado em [FRA 99] um estudo de caso sobre extração de conhecimento de documentos *SGML* para gerar uma ontologia para uma base de conhecimento. Os autores destacam que o fato da linguagem *SGML* ser fortemente estruturada ajudou bastante o processo de extração. A linguagem *XML* está formando a base para o desenvolvimento de documentos na *Web*, e o *DOM* (*Document Object Model*) [APP 98] provê uma *API* para facilitar o trabalho de projetistas e programadores quanto ao acesso, navegação e manipulação de documentos escritos em *HTML* ou *XML*.

O *DOM*, por ser orientado a objetos, suporta facilmente o trabalho com ontologias, visto que existe uma similaridade entre hierarquia de classes (da orientação a objetos) e árvore de taxonomia de conceitualizações (das ontologias). Como afirma [CHA 99] “o projeto orientado a objetos de sistemas de software depende de uma ontologia de domínio apropriada. Objetos, seus atributos e métodos, espelham aspectos

do domínio que são relevantes para a aplicação. Sistemas de objetos oriundos de uma competente análise de um domínio podem frequentemente ser reutilizados por um programa de aplicação diferente.”

Tanto na *Web* quanto nos ambientes de software em geral, o modelo orientado a objetos está presente na linguagem de programação *Java* da *SUN*, na tecnologia *.Net* da *Microsoft* e também nas duas principais tecnologias de objetos distribuídos: *DCOM* da *Microsoft* e *CORBA* do *OMG*. O *DOM* oferece uma interface para estas tecnologias manipularem os conteúdos *HTML* e *XML*, conforme ilustrado na Figura 4.6.

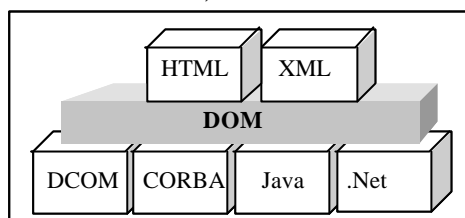


Figura 4.6: A Interface do *DOM*

Fonte: [BOR 99,p.36]

A versão do Assistente de *Feedback* neste estudo de caso não se comunica com outros agentes, todavia versões futuras podem incluir uma linguagem de comunicação e intercâmbio de conhecimento (tal como *KQML* [FIN 92] e/ou *FIPA ACL* [FIP 2004]). Neste caso, deve ser considerado que existem algumas dificuldades, como relata [LAB 99]: i) a tradução sintática entre linguagens; ii) a preservação do conteúdo semântico entre aplicações (mesmo se diferentes aplicações usam nomes diferentes para referenciá-lo); e, iii) a interação complexa entre agentes (onde existe o uso de estados de comportamento que representam atitudes mentais). A Figura 4.7 ilustra estes aspectos e mostra onde se aplicam ontologias no intercâmbio de conhecimento.

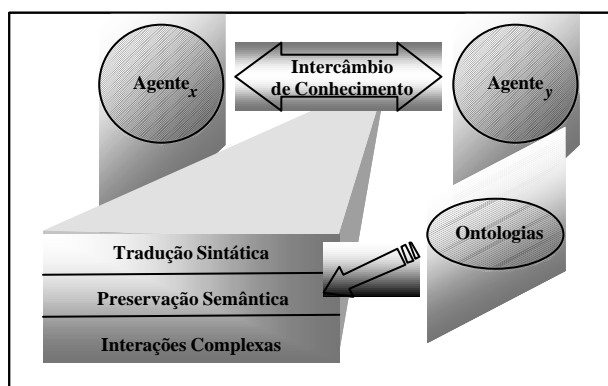


Figura 4.7: Ontologias na Comunicação entre Agentes de Software

Fonte: [BOR 99,p.37]

4.3 Serviço de Percepção do Ambiente do Agente de Software

O *Ambiente* (Definição 1.3, Cap.3) do agente de software Assistente de *Feedback* é um ambiente de trabalho de uma interface gráfica, onde o usuário usa as seguintes ferramentas/aplicações: Editor de Texto, Correio Eletrônico, Navegador *Web*, Gerenciador de Arquivos e Gerenciador de Agenda (*Scheduler*).

Neste Ambiente, os *Estímulos Internos* (em Definição 2.2.1, Cap.3) são os eventos de interação do usuário com a interface gráfica, e/ou as chamadas de serviços de software feitas pelas aplicações ou pelo sistema operacional. Vale lembrar que *Estímulos Internos* são *Percepções Certas* (Definição 2.2.1.7, Cap.3), isto é, são eventos

num contexto bem-definido, que provêm simultaneamente a sintaxe e a semântica). Quanto aos *Estímulos Externos* (em Definição 2.2.1,Cap.3), eles são *Percepções Incertas* (Definição 2.2.1.8, Cap.3), que necessitam de detecção de contexto (voltaremos a este ponto na Seção 4.4). Note que, a chegada de um *e-mail* (embora seja um evento que teve uma origem externa) é uma *Percepção Certa*, onde as ações do usuário com relação ao conteúdo deste *e-mail* também são *Percepções Certas*. Existe uma detecção de contexto inicial (que é a associação do Conteúdo Filtrado com o Contexto_de_Exclusão, e do Conteúdo não-Filtrado com o Contexto_de_Inclusão). Todavia, o mais importante e mais difícil envolve incerteza – que é inferir se o usuário possui ou não interesse no conteúdo relacionado com este *e-mail*. Eventualmente, futuras versões do Assistente de *Feedback* poderão tratar outros *Estímulos Externos*, tal como imagens da expressão facial do usuário e também comandos de voz do usuário, ou simplesmente expressões vocálicas do mesmo.

Os *Símbolos* (Definição 2.2.1.1, Cap.3) básicos são as palavras (ou Termos, siglas, *tokens*, etc). Os sensores que detectam estes *Símbolos* são programas *parsers* que analisam os Conteúdos (Filtrados e não-Filtrados). Conforme mostrado na Seção 4.2, a operação destes *parsers* é facilitada com o uso de padrões de representação de Conteúdo (por exemplo: *XML*, *RDF*).

Também fazem parte dos sensores os códigos de métodos, funções, macros (os quais estão embutidos nas ferramentas/aplicações e são feitos sob medida para capturar eventos específicos, por exemplo: quando o usuário salva um arquivo usando o editor de texto; ou, quando o usuário interage com um *e-mail*). Conforme também mostrado na Seção 4.2, esta captura de eventos é facilitada com o uso do *Document Object Model (DOM)*, o qual define uma *API (Application Programmers Interface)* para acesso, navegação e manipulação de documentos escritos em *HTML* ou *XML*.

Estes sensores fazem o *Serviço de Percepção do Ambiente* (Definição 2.2.1, Cap.3), coletando de forma contínua e incremental os eventos do Ambiente do Assistente de *Feedback*. Nesta versão do Assistente de *Feedback*, não é feito reconhecimento de padrões (de som, ou imagem, etc). Além da semântica embutida nos seus conteúdos (por exemplo, metadados *RDF* [LAS 99]), os *Símbolos* percebidos são ainda acrescidos da semântica dos seus *Contextos* (Definição 2.2.2.1,Cap.3) de origem no Ambiente do Assistente de *Feedback*, descritos a seguir:

- Contextos de Sugestão (podem ser considerados contextos de ações do Assistente de *Feedback*): são os Contexto_de_Exclusão e Contexto_de_Inclusão definidos na Seção 4.1;
- Contextos de Localidades: são os contextos do local onde os eventos estão sendo percebidos, ou seja, correspondem às ferramentas: Editor de Texto, Correio Eletrônico, Navegador *Web*, Gerenciador de Arquivos e Gerenciador de Agenda (*Scheduler*).
- Contextos de Funcionalidades (podem ser considerados contextos de ações do usuário): são contextos que acrescentam uma semântica que reflete a natureza da atividade do usuário. Na análise do Ambiente deste Assistente de *Feedback*, as funcionalidades básicas são:
 - Aprendizagem: É muito provável que o usuário esteja aprendendo um conteúdo (servindo como indicador de interesse neste conteúdo), quando ele *Anota_Texto*, e/ou quando ele *Lê_Texto*.

- Criação: É muito provável que o usuário esteja criando conteúdo (servindo como indicador de interesse neste conteúdo), quando ele *Compõe_Texto* (e/ou desenha figuras), e/ou quando ele *Define_Estruturas* - por exemplo, hierarquias de: diretórios/pastas no sistema de arquivos, pastas na caixa de *e-mail*, e/ou tópicos de endereços *Web* anotados (os “Favoritos”, ou os *Bookmarks*).
 - Eliminação: É muito provável que o usuário esteja eliminando conteúdo (servindo como indicador de desinteresse neste conteúdo), quando ele *Apaga_Arquivo*, *Apaga_Email*, *Apaga_Reunião_Agendada*, e/ou *Apaga_Tarefa_Agendada*.
 - Descoberta: É muito provável que o usuário tenha descoberto algo interessante, quando ele *Anota_Bookmark*, e/ou quando ele *Acessa_&Armazena* um novo **Objeto de Informação** - por exemplo: uma notícia filtrada pelo Serviço de Canais, e/ou uma página *Web*.
 - Seleção: É muito provável que o usuário esteja selecionando um conteúdo, quando ele cria filtros nos Canais de Informação, quando submete consultas às máquinas de busca, e/ou quando ele escolhe um *link* (dentre um conjunto de *links* numa página *Web*) enquanto “navega”.
 - Comunicação: É muito provável que o usuário esteja comunicando quando ele:
 - i) *Transporta_Conteúdo* (de *Objetos_de_Informação*, via *e-mail* ou *web*); e/ou
 - ii) *Interage_com_Pessoas*, por exemplo: via *e-mail*, ou reunião virtual via *web*, ou reunião real (registrando o evento no Gerenciador de Agenda).
 - Gerenciamento: É muito provável que o usuário esteja gerenciando suas atividades quando ele *Agenda_Reunião*, e/ou quando ele *Agenda_Tarefa*.
- Contextos de Atividades: são contextos que acrescentam uma semântica sobre o escopo da relação social envolvida nas atividades do usuário. Na análise do Ambiente deste Assistente de *Feedback*, as atividades podem ser de âmbito:
 - Profissional: todas as ações e tarefas do usuário relacionadas com o seu ambiente de trabalho. O usuário define para o Assistente de *Feedback* quais são as pessoas, tarefas e classes de documentos que identificam a atividade profissional.
 - Particular: todas as ações e tarefas do usuário relacionadas com a sua vida pessoal, familiar e/ou círculo de amizades. Tudo o que não foi definido para o Assistente de *Feedback* como atividade profissional é considerado atividade particular.
 - Contextos de Tempo: são contextos que delimitam os relacionamentos entre os eventos percebidos nas dimensões de Tempo Absoluto (Ano/Mês/Dia-Semana/Dia-Mês/Hora) e Tempo Lógico: *Tempo de Sistema* T_S (Definição 2.2.1.3, Cap.3) e *Tempo Contextual* T_C (Definição 2.2.2.2, Cap.3). O *Tempo Contextual* está estritamente relacionado com a semântica dos *Contextos* de origem dos eventos.

Por exemplo, no Ambiente do Assistente de *Feedback*, o usuário recebe uma mensagem M de *e-mail* de uma pessoa P e sobre um conteúdo C , num Tempo Absoluto ‘02/05/03’ e *Tempo Contextual* T_C , ou seja, $M(P,C, '02/05/03', T_C)$. Ao receber M , o usuário realizou as ações de *Ler_Título* e *Ler_Corpo*, ou seja:

- $M(P,C, '02/05/03', T_C) \leftarrow \text{Ler_Título};$
- $M(P,C, '02/05/03', T_C) \leftarrow \text{Ler_Corpo}.$

Depois disso, o usuário ficou um longo período de Tempo Absoluto, interagindo (via-email) com outras pessoas e sobre diversos outros assuntos. Todavia, na dimensão de *Tempo Contextual*, a mensagem $M(P,C,02/05/03,T_C)$ continua “vivendo um tempo presente contínuo”. Após o referido período de Tempo Absoluto, este usuário terminou de editar um documento D sobre o conteúdo C , então fez uma pesquisa (casando o conteúdo C) na Ferramenta de Correio Eletrônico para selecionar/recuperar M . Enfim, ao acessar M , o usuário faz a ação $\text{Responde_Msg}(M)$, incluindo como anexo o documento D . Portanto, na dimensão de *Tempo Contextual* (e na perspectiva) da mensagem M ocorreu o seguinte:

- $M(P,C, '02/05/03', T_C) \leftarrow \text{Ler_Título}$;
- $M(P,C, '02/05/03', T_C) \leftarrow \text{Ler_Corpo}$;
- $M(P,C, '12/09/03', T_{C+1}) \leftarrow \text{Pesquisa_Msg}(C)$;
- $M(P,C, '12/09/03', T_{C+1}) \leftarrow \text{Abre_Msg}$;
- $M(P,C, '12/09/03', T_{C+1}) \leftarrow \text{Responde_Msg\&Anexa}(D)$.

Uma analogia que se pode fazer é comparar o Tempo Absoluto com a visão de uma janela de mensagens de *e-mail* ordenadas por data de recebimento. Já o *Tempo Contextual* tem uma certa similaridade com uma janela de mensagens de *e-mail* organizada em *threads* (onde uma mensagem M_1 é seguida por outra $M_{1.1}$ que responde a M_1 e assim por diante). Em outras palavras, é como se no *Tempo Contextual*, cada evento relacionado (e sucessivo) fosse uma “resposta” para o evento anterior, isto é, o evento anterior pode ser a causa e o evento posterior pode ser o efeito (mas ao mesmo tempo em que um evento pode ser efeito de um evento anterior, ele pode também causar o próximo).

Para entender melhor os *Relacionamentos Semânticos Multidimensionais* (Definição 2.3.1, Cap.3) que existem entre os *Contextos* no *Ambiente* do Assistente de *Feedback*, a Figura 4.8 mostra as relações entre os *Contextos*: Localidades, Funcionalidades, e Sugestão (*Contexto_de_Exclusão* e *Contexto_de_Inclusão*). Nesta Figura 4.8, para facilitar a visualização, o eixo principal dos relacionamentos multidimensionais é representado por uma estrutura de árvore das Funcionalidades, onde nas folhas estão as Localidades: (E) Editor, (G) Gerenciador de Arquivos, (C) Correio Eletrônico, (N) Navegador, (A) Agenda/*Scheduler*. Ainda para facilitar a visualização, as Localidades estão repetidas nas folhas – mas deve ser considerado que uma única Localidade se relaciona com várias Funcionalidades, por exemplo: (E) Editor está relacionado com *Aprendizagem/Anota_Texto*, *Aprendizagem/Lê_Texto*, *Criação/Compõe_Texto* e *Seleção/Consulta_de_Pesquisa*.

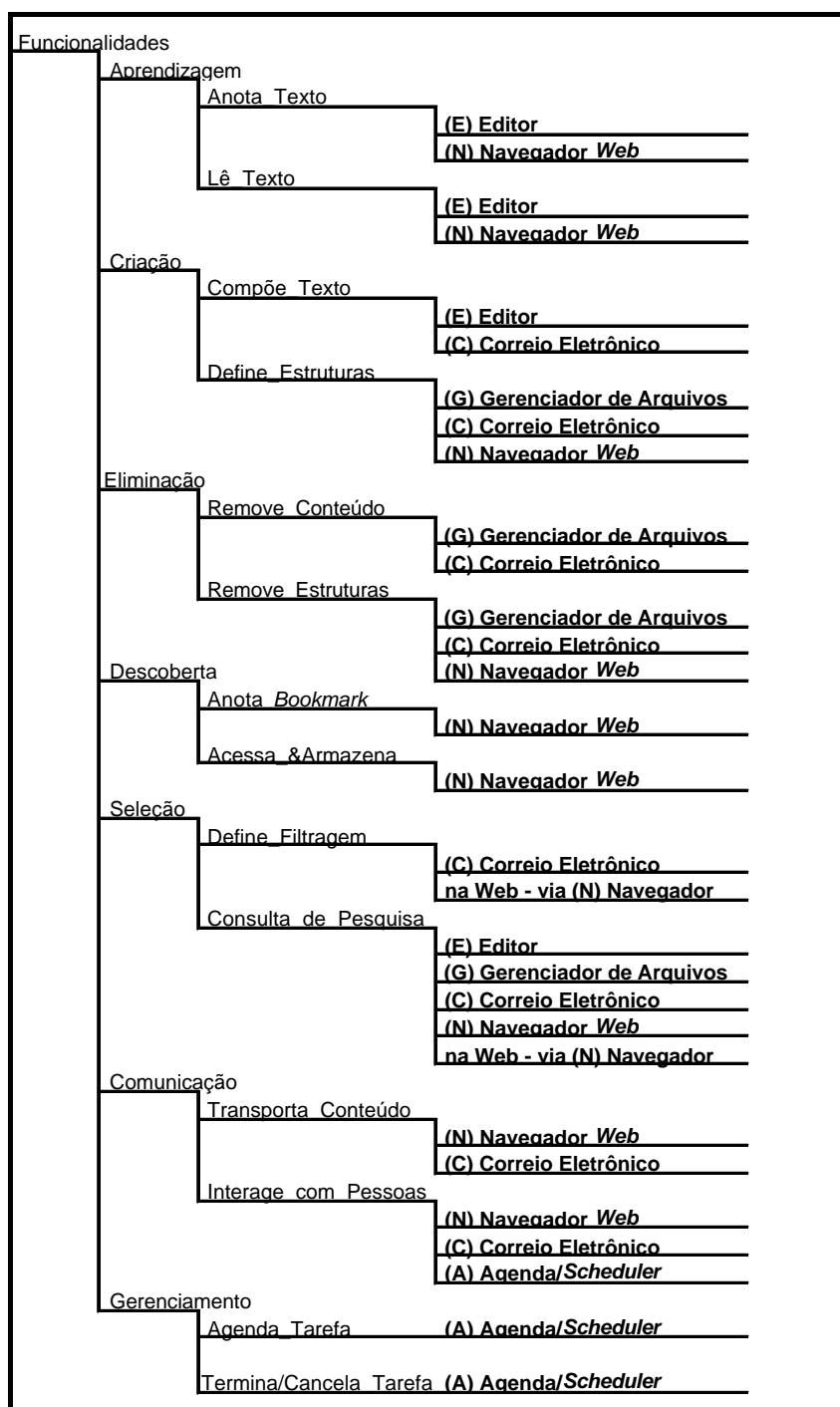


Figura 4.8: Relacionamento entre os *Contextos*: Localidades, Funcionalidades e Sugestão (Exclusão e Inclusão)

Os Símbolos percebidos, juntamente com os seus *Contextos* de origem, fazem parte do *Espaço de Contexto* (Definição 2.2.2.3, Cap.3) multidimensional do Assistente de *Feedback*. Este espaço é primeiramente particionado pelos *Contextos* de Localidades, Funcionalidades, Atividades e Tempo. Depois, cada partição do espaço é subdividida em dois subespaços formados pelos *Contexto_de_Exclusão* e *Contextos_de_Inclusão*. Por último, cada subespaço é preenchido pelos *Símbolos* percebidos. A Figura 4.9 ilustra o particionamento do *Espaço de Contexto*.

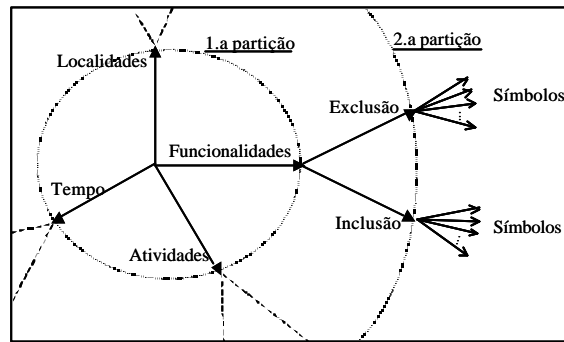


Figura 4.9: Particionamento do *Espaço de Contexto* multidimensional do Assistente

Para exemplificar uma instância neste *Espaço de Contexto* multidimensional do Assistente de *Feedback*, considere o evento do usuário apagar uma mensagem M de um amigo A_n convidando para uma partida P_x de futebol, ou seja, $M(A_n, P_x)$. A Figura 4.10 ilustra a formação do *Símbolo Composto* (Definição 2.2.1.2, Cap.3) deste evento, onde o fato de M ser um Conteúdo não-Filtrado faz com que ela pertença ao Contexto_de_Inclusão.

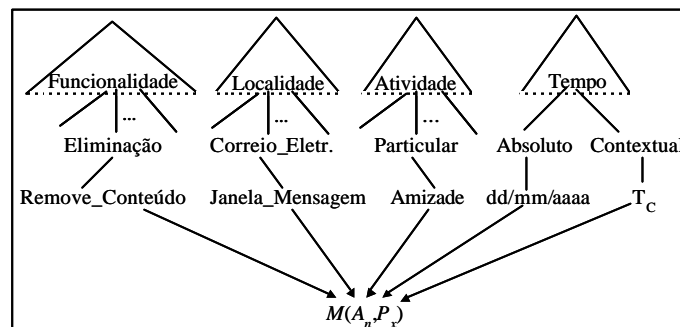


Figura 4.10: Exemplo de *Símbolo Composto* no Assistente de *Feedback*

Após exemplificar a formação de um *Símbolo Composto*, vamos ilustrar agora como ocorre o *Fluxo de Percepção* (Definição 2.2.1.6, Cap.3) onde um evento (na forma de *Símbolo Composto*) é distribuído para vários *Contextos*. Vamos considerar aquele primeiro exemplo onde o usuário recuperou (fazendo uma pesquisa) um *e-mail* antigo, depois respondeu este *e-mail* anexando nele o documento D . Mais especificamente, vamos considerar um daqueles eventos ocorridos no *Tempo Contextual* mais recente, isto é, T_{C+1} :

- $M(P, C, '12/09/03', T_{C+1}) \leftarrow \text{Pesquisa_Msg}(C)$;

Este evento, para o Assistente de *Feedback*, corresponde à *Percepção da Ação* do usuário $PA_{T_{C+1}} = \text{Pesquisa_Msg}(C)$ relacionada com a *Percepção de Objeto do Ambiente* $PO_{T_{C+1}} = M(P, C, '12/09/03', T_{C+1})$. (Obs.: *PA* e *PO* da Definição 2.2.1.5, Cap.3).

Com relação à *Pesquisa_Msg(C)*, ela ocorreu no Contexto de Localidade 'Ferramenta de Correio Eletrônico'. Todavia, na dimensão do Contexto de Funcionalidades, a função de pesquisar por um conteúdo C ocorre também em outros Contextos de Localidade, portanto *Pesquisa(C)* é distribuído para vários Contextos de Localidade. Na aplicação deste estudo de caso, a distribuição de '*Evento(Conteúdo)*' fornece indicadores de preferências de conteúdo (sendo que a coleta de indicadores subsidia o processo de sugestão).

Note que cada Contexto de Localidade vai tratar o evento de uma forma muito específica, dependendo da semântica do Contexto, e do histórico de relações causais que este evento possui no Contexto. A Figura 4.11 ilustra a distribuição do evento *Pesquisa(C)*, tendo como origem a Localidade ‘Ferramenta de Correio Eletrônico’ e como destino as outras Localidades, sendo que esta distribuição se propagou pela dimensão do Contexto de Funcionalidades. Ou seja, uma ação de *Pesquisa* sobre o conteúdo *C* fornece uma indicação sobre o foco da atenção/interesse do usuário, subsidiando a formação de Conteúdo Candidato (a ser descrito na Seção 4.6.1).

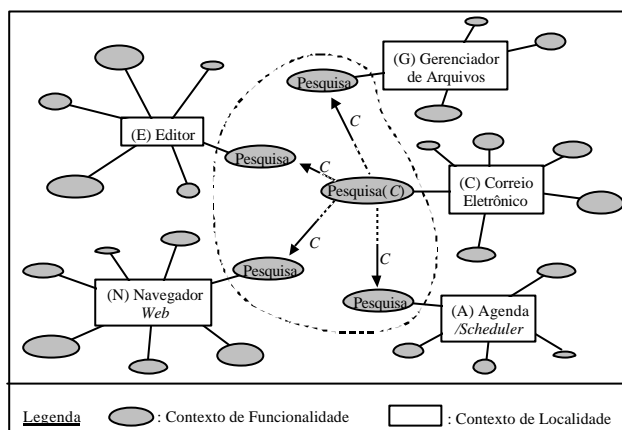


Figura 4.11: Exemplo de Distribuição de Evento no Assistente de *Feedback*

Os principais eventos do *Ambiente* do Assistente de *Feedback* são apresentados nas Figuras 4.12 e 4.13, onde os eventos estão reunidos por Localidades. Em cada localidade, os eventos fornecem indicadores de interesse (+) e desinteresse (-) pelo conteúdo *C*, cujos valores são baseados na análise do domínio feita por especialistas humanos (assim como em [HEC 98]). Neste estudo de caso, estes valores são definidos de forma empírica. Alguns eventos possuem mais de um valor para o indicador de interesse (p.ex. ‘Editor.Abre_Documento’), que vai depender do contexto (isto é descrito na Subseção 4.4.3). Os eventos que iniciam com ‘AF_’ são eventos do Assistente de *Feedback*. Os eventos ‘AF_Detecta_’ são eventos que reagem imediatamente às ações do usuário (e este não os percebe). Já os eventos ‘AF_Sugere_’ ocorrem após um processo de inferência (e o usuário, além de perceber, também provê *Feedback*).

Editor		
Abre_Documento (+1, 2 ou 3)	Inclui_Documento (+1, 2 ou 3)	AF_Detecta_Autores
Pesquisa_no_Texto (+1)	Cola_Texto (+1 ou 2)	AF_Detecta_Conteúdo
Copia_Texto (+1 ou 2)	Imprime_Texto (+1)	AF_Detecta_Destino_Arquivo
Lê_Texto (+1)	Anota_Texto (+3)	AF_Detecta_Data
Salva_Texto (+3, 10 ou 25)		AF_Sugere_Anotação
Digita_Texto (+1)		AF_Sugere_Leitura
Navegador Web		
Acessa_Página (+1)	Estrutura_Bookmarks (+3)	AF_Detecta_Autores
Anota_Texto (+1)	Pesquisa_na_Web (+1)	AF_Detecta_Conteúdo
Copia_Texto (+1)	Cadastra_Canal_Web (+25)	AF_Detecta_Destino_Arquivo
Cola_Texto (+1)	Cadastra_Filtro_Canal (+25)	AF_Detecta_Origem_Web
Lê_Texto (+1)	Envia_Texto (+2)	AF_Detecta_Data
Imprime_Texto (+1)	Recebe_Texto (+2)	AF_Sugere_Pesquisa
Pesquisa_Texto (+1)		AF_Sugere_Anotação
Salva_Texto (+2 ou 5)		AF_Sugere_Leitura
Anota_Bookmark (+3)		AF_Sugere_Canal_Web
		AF_Sugere_Filtro_Canal

Figura 4.12: Eventos Percebidos no Editor e Navegador *Web*

Correio Eletrônico		
Visualiza_Msgs (+1)	Encaminha_Msg (+2)	AF_Detecta_emails_Pessoas
Lê_Título_Msg (+1)	Salva_Texto_Msg (+2 ou 5)	AF_Detecta_Conteúdo
Lê_Corpo_Msg (+2)	Salva_Anexo_Msg (+2 ou 5)	AF_Detecta_Destino_Anexo
Digita_Msg (+5)	Imprime_Msg (+1)	AF_Detecta_Origem_Web
Copia_Texto (+1)	Pesquisa_Msg (+2)	AF_Detecta_Destino_Web
Cola_Texto (+1)	Remove_Msg (-3)	AF_Detecta_Data
Inclui_Anexo_Msg (+5)	Registra_email_Pessoa (+10)	AF_Sugere_Anexo
Responde_Remetente (+3)	Cadastra_Filtro_Msg (+15)	AF_Sugere_email_Pessoa
Responde_paraTodos (+3)	Estrutura_Pastas_Inbox (+3)	
Agenda		
Visualiza_Tarefas (+1)	Inclui_Documento (+5)	AF_Detecta_Pessoas
Inserere_Tarefa (+10)	Cancela_Tarefa (-10)	AF_Detecta_Conteúdo
Copia_Texto (+1)	Pesquisa_Tarefa (+1)	AF_Detecta_Prazo_Tarefa
Cola_Texto (+1)	Executa_Tarefa (+10)	AF_Detecta_Local
Lê_Tarefa (+1)		
Gerenciador de Arquivos		
Visualiza_Arquivos (+1)	Recorta_Arq (-10 ou 20)	AF_Detecta_Autores
Estrutura_Pastas_Arq (+5)	Salva_Arq (+2 ou 5)	AF_Detecta_Conteúdo
Apaga_Arq (-10, 50 ou 100)	Pesquisa_Nome_Arq (+2)	AF_Detecta_Destino_Arquivo
Copia_Arq (+1)	Pesquisa_Conteúdo_Arq (+3)	AF_Detecta_Data
Cola_Arq (+1 ou 2)		

Figura 4.13: Eventos Percebidos no Correio, Agenda e Gerenciador de Arquivos

Os eventos das Figuras 4.12 e 4.13 que não iniciam com ‘AF_’ são *Percepções de Ações* do usuário, isto é (PA). Estas Ações são realizadas sobre Objetos de Informação do Ambiente, os quais são alvos das *Percepções de Objetos* (PO). A Figura 4.14 mostra os principais objetos que representam as *Percepções de Objetos do Ambiente* feitas pelo Assistente de *Feedback* (note que estes objetos representam as percepções (PO), e não os Objetos de Informação propriamente).

Documento	Título Palavras-Chave Autor(es) Data Endereço_Pasta_Arquivo	Correio Eletrônico	Remetente Destinatário(s) Assunto Palavras-Chave Data Anexo	Tarefa Agenda	Assunto Tarefa* Prazo Pessoas
Página Web	Título Palavras-Chave Autor(es) Data Endereço_Web Endereço_Local	Reunião Agenda	Assunto Local Data Pessoas	*Tarefa: qualquer item do Contexto de Funcionalidades.	

Figura 4.14: Representação dos objetos percebidos pelo Assistente de *Feedback*

Assim como os eventos percebidos de Ações do Usuário (Figuras 4.12 e 4.13) estão relacionados com uma hierarquia de classes de Funcionalidades (Figura 4.8), os atributos de objetos percebidos pelo Assistente de *Feedback* também estão relacionados com uma hierarquia de *Classes que Correspondem a Estruturas de Linguagem (CCEL)* (em Definição 2.2.2.6, Cap.3). A Figura 4.15 mostra estes relacionamentos.

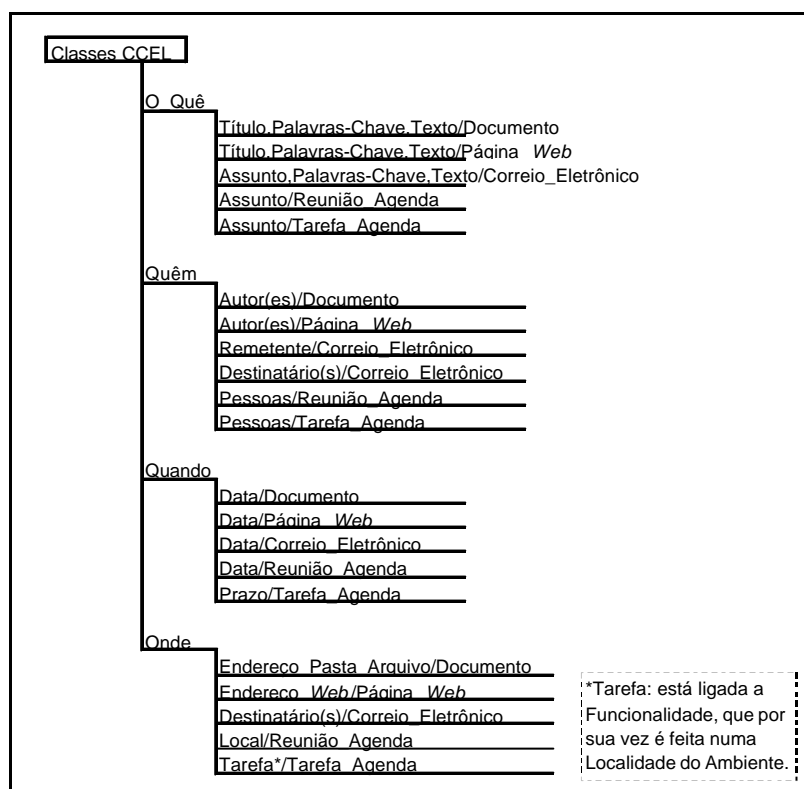


Figura 4.15: Relação dos atributos de objetos percebidos com as Classes CCEL

Com relação à *Entrada de Percepções* (Definição 2.2.1.4, Cap.3) de Objetos (PO), Ações (PA) e *Feedback Interior* (PI), o Assistente de *Feedback* percebe dois tipos de {PO,PA,PI}: 1^a) As Ações do usuário relacionadas com os Objetos do Ambiente deste usuário; e, 2^a) As Ações de Sugestão do Assistente de *Feedback* relacionadas com 'As Ações do usuário relacionadas com os Objetos do Ambiente deste usuário'. A Entrada de Percepções é feita de forma contínua e incremental. A Figura 4.16 ilustra a coleta de eventos através do *Serviço de Percepção do Ambiente* (Definição 2.2.1, Cap.3), onde cada seta simboliza o serviço de percepção realizado pelos *parsers*, métodos, funções e macros que estão embutidos nas ferramentas/aplicações.

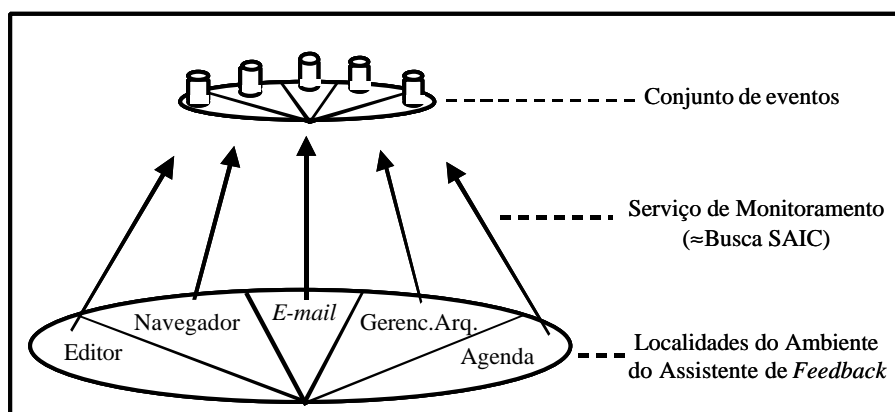


Figura 4.16: Coleta de eventos no Ambiente do Assistente de *Feedback*

4.4 Serviço de Detecção de Contexto do Agente de Software

Ao longo deste trabalho a importância dos contextos como provedores de semântica foi bastante enfatizada, e na seção anterior foram descritos os *Contextos* do Ambiente do Assistente de *Feedback*. Conforme descrito no *Serviço de Detecção de Contexto* (Definição 2.2.2, Cap.3), é feito um casamento (*matching*) entre a sintaxe e a semântica dos eventos/Símbolos percebidos, usando a *Entrada de Percepção* (provida pelo *Serviço de Percepção do Ambiente*) juntamente com a leitura do estado de ação ativo do *Modelo de Mundo* (Definição 2.2.2.4, Cap.3).

Portanto, esta seção inicia descrevendo a *Entrada de Percepção* no Ambiente do Assistente de *Feedback* e o *Serviço de Detecção de Contexto* (Subseção 4.17), para depois descrever o *Modelo de Mundo* do Assistente de *Feedback* (Subseção 4.4.2) e concluir descrevendo os relacionamentos entre a *Entrada de Percepção* e o *Modelo de Mundo* (Subseção 4.4.3).

4.4.1 Entrada de Percepção e o Serviço de Detecção de Contexto

O *Serviço de Detecção de Contexto* normalmente ocorre de forma incremental, pois o Ambiente do Assistente de *Feedback* também possui a propriedade de *Mudanças Incrementais* (em Definição 2.2.2.3, Cap.3): por ser um Ambiente virtual planejado, o usuário realiza mudanças incrementais no *Espaço de Contexto* (Definição 2.2.2.3, Cap.3), porque cada transição de um lugar para outro é feita sequencialmente, passo-a-passo (isto é, clique-a-clique). Adicionalmente, estes lugares interligados estão semanticamente próximos no *Espaço de Contexto*. Por exemplo, quando o usuário abre uma interface de uma aplicação, os *links* e botões exibidos nesta interface – sempre que ativados – conduzirão o usuário para um lugar cuja funcionalidade é compatível com o propósito da aplicação. Todavia, eventualmente o Assistente de *Feedback* tem que fazer uma *Detecção Massiva de Contexto* (em Definição 2.2.2.3, Cap.3) enquanto o sistema operacional do computador está carregando a configuração do ambiente de trabalho (*workspace*) de uma nova sessão (*login*) de um usuário; ou, quando ocorrer alguma falha (de software, hardware, ou falta de energia elétrica).

No Ambiente do Assistente de *Feedback*, mais especificamente, existe uma relação entre o tipo da percepção e a necessidade de Detecção de Contexto. Neste Ambiente, as ações do usuário são *Percepções Certas* (por exemplo, um clique de *mouse* sobre um *link* ou botão). A Percepção Certa possui uma semântica certa/específica que foi definida pelo projetista do Ambiente do software. Portanto, o Assistente de *Feedback* percebe (sem ambigüidade) o que o usuário está fazendo. Apesar do Assistente de *Feedback* coletar *Percepções Certas* sobre as ações do usuário, ele as trata como *Percepções Incertas* – nos seguintes aspectos de incerteza: a) quais são as preferências (e/ou não-preferências) do usuário; e, b) qual é a duração de cada preferência (e/ou não-preferência). Portanto, os eventos/Símbolos que são *Percepções Incertas* precisam do *Serviço de Detecção de Contexto* para associar a semântica.

Nesta subseção, é considerado que a *Entrada de Percepção* contém as *Percepções Certas*, pois o Ambiente do Assistente de *Feedback* fornece os eventos com sintaxe e semântica bem definidos (como se houvesse uma pré-Detecção de Contexto). Quanto ao *Serviço de Detecção de Contexto* para as *Percepções Incertas* (as preferências do usuário e duração destas), voltaremos a considerá-las agora na Subseção 4.4.2, ao descrever o Comportamento do Assistente de *Feedback* (incluído no *Modelo de Mundo*).

Conforme descrito no *Framework SADI*, o *Espaço de Percepção* possui três subespaços principais que possuem um relacionamento especial (PO,PA,PI) para dar suporte às operações incrementais deste *Framework*. Assim, no Assistente de *Feedback*, o *Espaço de Percepção* também possui subespaços (PO,PA,PI). Quanto ao subespaço *PA* (*Percepção de Ações*), ele está organizado na hierarquia do Contexto de Funcionalidades (Figura 4.8), sendo que esta hierarquia classifica os eventos das Ações (Figuras 4.12 e 4.13).

O Assistente de *Feedback* é uma aplicação do *Framework SADI*, o qual é baseado no *Framework SAIC*. Considerando que os eventos que correspondem à *Percepção de Ações* já estão sendo coletados continuamente (Figura 4.16), então o serviço de Filtragem SAIC (Definição 2.1.2, Cap.3) é usado para organizar (com operações contínuas e custos incrementais) o subespaço *PA*, conforme ilustra a Figura 4.17). Na base da Figura 4.17 está representado o Conjunto de eventos coletados, onde cada Contexto de Funcionalidade (Aprendizagem, Eliminação, Criação, Seleção, Comunicação e Gerenciamento) cadastra um filtro em cada Contexto de Localidade. No topo da Figura 4.17 estão representadas as classes de ações de cada Contexto de Funcionalidade, sendo que cada classe de ação cadastra um filtro no seu respectivo Contexto de Funcionalidade (por exemplo, na Funcionalidade de Aprendizagem estão os filtros das classes de ação: *Anota_Texto* e *Lê_Texto*).

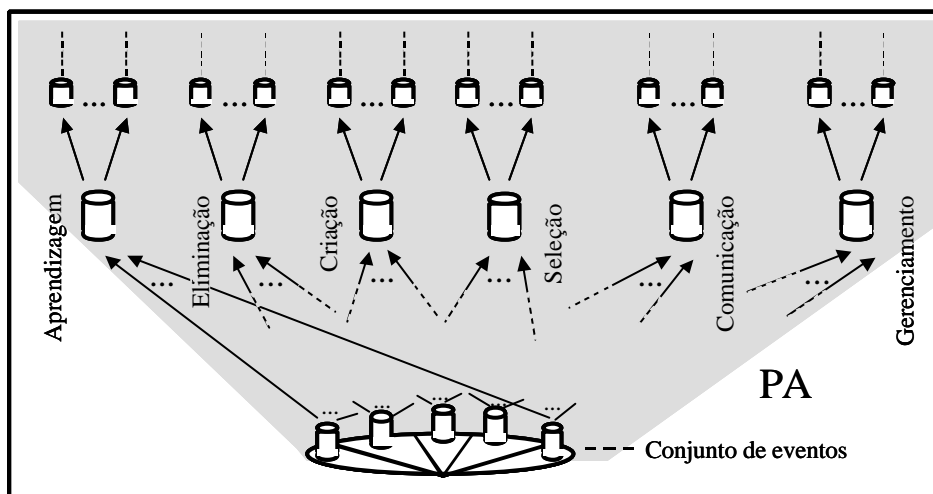


Figura 4.17: Organização do Subespaço de Percepção de Ações (PA)

Quanto ao subespaço *PO* (*Percepção de Objetos*), ele está organizado na hierarquia de *Classes que Correspondem a Estruturas de Linguagem (CEEL)* (Figura 4.15), sendo que esta hierarquia classifica os atributos dos objetos percebidos pelo Assistente de *Feedback* (Figura 4.14). Da mesma forma que o serviço de Filtragem SAIC organiza o subespaço *PA* (na Figura 4.17), ele também é usado para organizar o a hierarquia do subespaço *PO*. A Figura 4.18 simboliza como as hierarquias dos subespaços *PO* e *PA* do Assistente de *Feedback* são construídas pelos *Serviços SADI/SAIC*. A percepção simultânea de *PO* e *PA* formam os Símbolos Compostos ($PO_{T_c}.PA_{T_c}$) que ficam esperando pelo *Fluxo de Feedback* em T_{C+1} para formar o cubo que representa na Figura 4.18 a formação dos relacionamentos ($(PO_{T_c}.PA_{T_c})PI_{T_c+1}$).

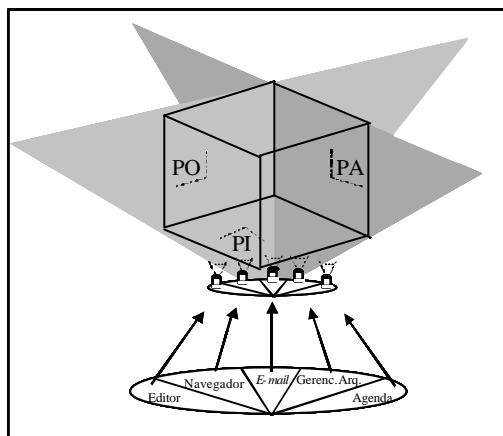


Figura 4.18: Construção dos subespaços PO, PA e PI do Assistente de *Feedback*

Na Definição 2.3 (Capítulo 3), é mostrado que a sobreposição de *Símbolos Compostos* comuns estabelece três relacionamentos simultâneos e úteis (1^o Relacionamentos Causais; 2^o Relacionamentos Semânticos Multidimensionais; e, 3^o Estruturas de Acesso aos Dados). Sobre o 2^o relacionamento, apesar da topologia hierárquica do roteamento por conteúdo (Filtragem SAIC) dos *Símbolos* percebidos, a estrutura completa dos relacionamentos é multidimensional, onde o cubo básico é formado pelos subespaços (PO,PA,PI).

Em cada cubo básico, os subespaços (PO,PA,PI) representam o relacionamento dos Contextos de Funcionalidades, Localidades e Tempo (descritos na Seção 4.3). Quanto à organização geral dos cubos no espaço multidimensional, ela pode ser feita considerando: 1^o) Os Contextos de Sugestão (conforme o objetivo do Assistente de *Feedback*, Seção 4.1), compostos pelo Contexto de Exclusão (relacionado com o Conteúdo Filtrado) e Contexto de Inclusão (relacionado com o Conteúdo não-Filtrado); e, 2^o) Os Contextos de Atividades (Profissional e Particular, Seção 4.3). A Figura 4.19 ilustra esta organização geral dos cubos.

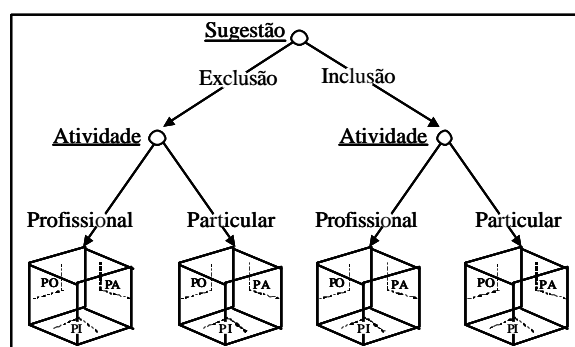


Figura 4.19: Organização dos cubos da estrutura multidimensional

Nesta subseção foi mostrado como são organizadas as *Percepções Certas* no Ambiente do Assistente de *Feedback*, as quais já trazem consigo a sintaxe e a semântica que identificam seus Contextos de origem. Mas, com relação às *Percepções Incertas* (as preferências do usuário e a duração destas), o Serviço de Detecção de Contexto faz a associação semântica da *Percepção Incerta* usando a *Entrada de Percepção* juntamente com o *Modelo de Mundo*, a ser descrito a seguir.

4.4.2 Modelo de Mundo do Assistente de *Feedback*

Uma vez que o objetivo do Assistente de *Feedback* é ajudar o usuário a atualizar seus Filtros nos Serviços de Canais de Informação, o *Modelo de Mundo* (Definição 2.2.2.4, Cap.3) do Assistente de *Feedback* deve refletir o comportamento do usuário (ou seja, deve mimetizar os padrões de comportamento do usuário). Através da observação das ações do usuário relacionadas com conteúdos (Filtrados e não-Filtrados), o Assistente de *Feedback* deve simular no *Modelo de Mundo* o comportamento do usuário. Portanto, as sugestões do Assistente de *Feedback* são baseadas nas ações mais prováveis que o usuário faria no modelo simulado. Basicamente, a contribuição do Assistente de *Feedback* decorre da sua maior capacidade de armazenamento e de organização contextual, visto que o usuário humano certamente encontra dificuldades ao lidar com sobrecarga de informação.

Portanto, o *Modelo de Mundo* do Assistente de *Feedback* inclui o Comportamento do Usuário e o Comportamento Próprio (do Assistente), descritos a seguir.

Comportamento do Usuário

Ao observar o mundo real, o trabalho de assistência ou de secretaria precisa de treinamento inicial e depois uma aprendizagem contínua sobre as necessidades e/ou preferências da pessoa que está sendo ajudada/assistida.

O Assistente de *Feedback*, também precisa conhecer quem está sendo ajudado. Numa versão desenvolvida para um ambiente de produção, o Assistente pode eventualmente oferecer um conjunto de perfis de trabalho – onde o usuário seleciona (e depois o Assistente de *Feedback* ajusta continuamente) o perfil de trabalho que corresponde ao seu papel funcional e *workflow* (idealmente, o Assistente de *Feedback* opera num Ambiente de software que possui um sistema de *workflow*, o qual provê ainda mais semântica sobre o comportamento do usuário).

Aqui nesta modelagem, será considerado somente o Contexto de Atividade Profissional, e apenas o perfil de trabalho de alguém que realiza pesquisa (de tendências científicas, de mercado, culturais, etc) e escreve relatórios contendo análises. Mais especificamente, este usuário U_1 recebe uma tarefa contendo um tema de pesquisa e um prazo para entregar o relatório. Para realizar esta tarefa, o usuário U_1 faz pesquisa pela Internet, realiza estudos sobre o material que encontrou na pesquisa, compõe/redige o relatório, e durante toda a tarefa interage/comunica com seu grupo de trabalho.

A Figura 4.20 ilustra o modelo básico do comportamento de um usuário na execução de sua tarefa. Nesta figura, o usuário U_1 exibe um padrão de comportamento no Contexto de Funcionalidades, sendo que Inicia e Termina Tarefa são eventos que ocorrem na Funcionalidade de Gerenciamento. Para simplificar a visualização da Figura 4.20, não está representada a Funcionalidade de Eliminação (de conteúdo e/ou estrutura), embora em qualquer momento o usuário U_1 possa executá-la.

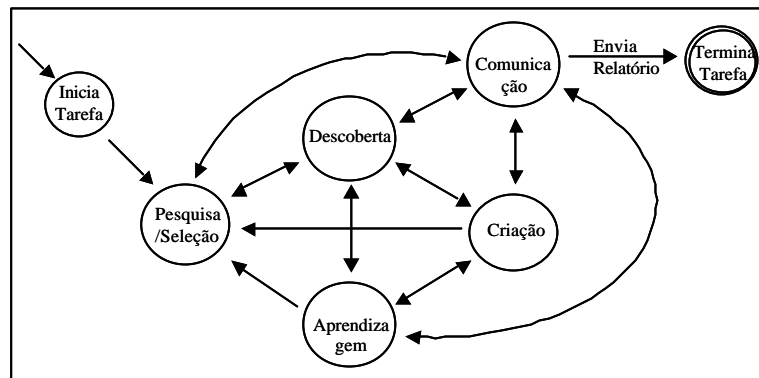


Figura 4.20: Exemplo de Modelo de Comportamento do Usuário

Todavia, cada usuário, ainda tendo o mesmo perfil de atividade (e *workflow*), vai estabelecer o seu próprio padrão de comportamento – por exemplo, considere um outro usuário U_2 , o qual tem o hábito de geralmente iniciar uma sessão de trabalho lendo *e-mails* (Comunicação), depois estuda o material pesquisado (Aprendizagem), em seguida redige algum texto (Criação), etc. A modelagem destes padrões de comportamento pode ser feita considerando as ligações criadas na rede que mapeia o comportamento e considerando também a frequência com que estas ligações são percorridas, assim como é feito também no Modelo Conexcionista para Hipermídia Adaptativa [PAL 2000].

Portanto, cada usuário possui o seu modelo de comportamento no Contexto de Funcionalidades, tal como ilustrado na Figura 4.20. Mas, conforme apresentado na Figura 4.8, cada Funcionalidade é composta por eventos relacionados com um ou mais Localidades (Editor de Texto, Correio Eletrônico, Navegador *Web*, Gerenciador de Arquivos e Gerenciador de Agenda). Ou seja, o modelo da Figura 4.20 corresponde a um primeiro nível, sendo que os eventos correspondem a um segundo nível, tal como ilustra a Figura 4.21.

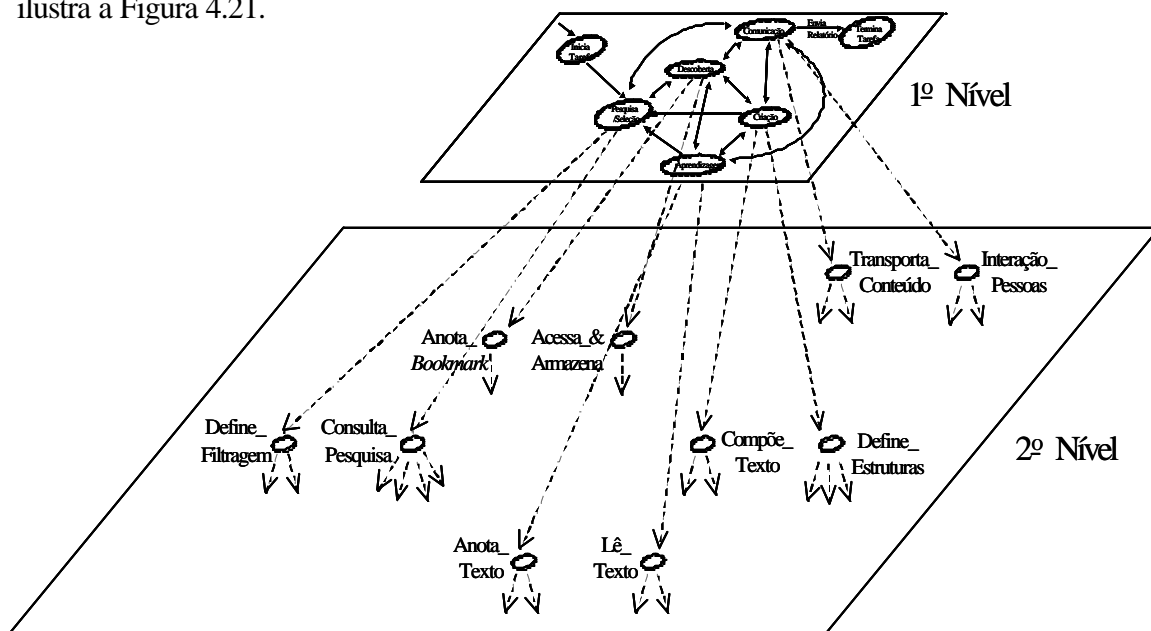


Figura 4.21: Exemplo dos Níveis do Modelo de Comportamento do Usuário

Quanto às ligações que saem do 2º nível (o plano inferior) da Figura 4.21, elas se dirigem às diferentes Localidades (aplicações/ferramentas) do Ambiente do Assistente de *Feedback*. Embora o *Modelo de Mundo* esteja focalizado na *Percepção das Ações* (PA) do usuário, é importante ter em mente que estas ações estão sempre relacionadas com um conteúdo de *Percepção de Objetos* (de Informação) do Ambiente (PO), conforme está representado na Figura 4.18.

A escolha deste enfoque baseado em ações se deve ao fato da estrutura do padrão de comportamento ser mais regular e estável do que o enfoque baseado no conteúdo (o qual varia muito mais dinamicamente). Note que este mesmo princípio de escolher padrões mais regulares e estáveis está na base da vantagem do Serviço de Filtragem SAIC (focalizada na estrutura dos subespaços de informação), ao invés de técnicas de consultas de pesquisa (focalizadas no conteúdo que preenche os subespaços de informação), conforme exposto no Teorema 2.1.1, Cap.3.

Além do Comportamento do Usuário (representado por Fluxos de Ações), em algumas Localidades são definidos os Locais de Sugestão do Assistente de *Feedback* e também são identificados os Fluxos de Conteúdo Rastreado. Este rastreamento é indicado pelas setas largas em cor cinza, que significam o rastreamento de conteúdo na dimensão de *Tempo Contextual* (vide exemplo do conteúdo *C* nas mensagens *M*, Seção 4.3). Uma vez que o objetivo do Assistente de *Feedback* é sugerir a remoção (e/ou inclusão) de Termos nos Filtros dos Canais de Informação (Seção 4.1), então o rastreamento dos Fluxos de Conteúdo serve para propagar semântica e auxiliar o processo de aprendizagem (a ser descrito na Seção 4.5).

Além disso, com relação à representação de probabilidades na Representação de Conhecimento SADI (Definição 2.3.1, Cap.3), por onde os fluxos de percepções (das ações) passam ao longo do modelo, eles atualizam os valores das probabilidades (e probabilidades condicionais) dos *Símbolos* percebidos. Mais adiante, na Seção 4.6, será mostrado como estas probabilidades são usadas pelo *Serviço de Tomada de Decisão* do Assistente de *Feedback* para fazer sugestões.

Para facilitar a exposição do Modelo do Comportamento do Usuário, continuaremos a detalhá-lo por Localidade:

- *Navegador Web*. A Figura 4.22 ilustra um exemplo de *Modelo de Mundo* do comportamento do usuário usando o *Navegador Web*. Nesta figura, cada círculo representa um estado de ação do usuário (que foi percebido pelo Assistente), e as setas indicam os sentidos dos fluxos das ações do usuário. Quanto à representação de probabilidades, estes valores são atualizados por onde os fluxos de percepções passam. Por exemplo, se o usuário acessou uma página *Web*, leu seu texto e depois fez anotações sobre o texto lido, então são atualizadas as probabilidades $P(\text{Acessa_Página})$, $P(\text{Lê_Texto}|\text{Acessa_Página})$ e $P(\text{Anota_Texto}|\text{Acessa_Página}, \text{Lê_Texto})$. Aqui, o Assistente de *Feedback* pode fazer sugestão quando o usuário faz pesquisa na web (submetendo consultas para uma máquina de busca), quando o usuário usa uma ferramenta de anotar texto, ou quando o usuário acessa o serviço de Filtragem de Canais de Informação.

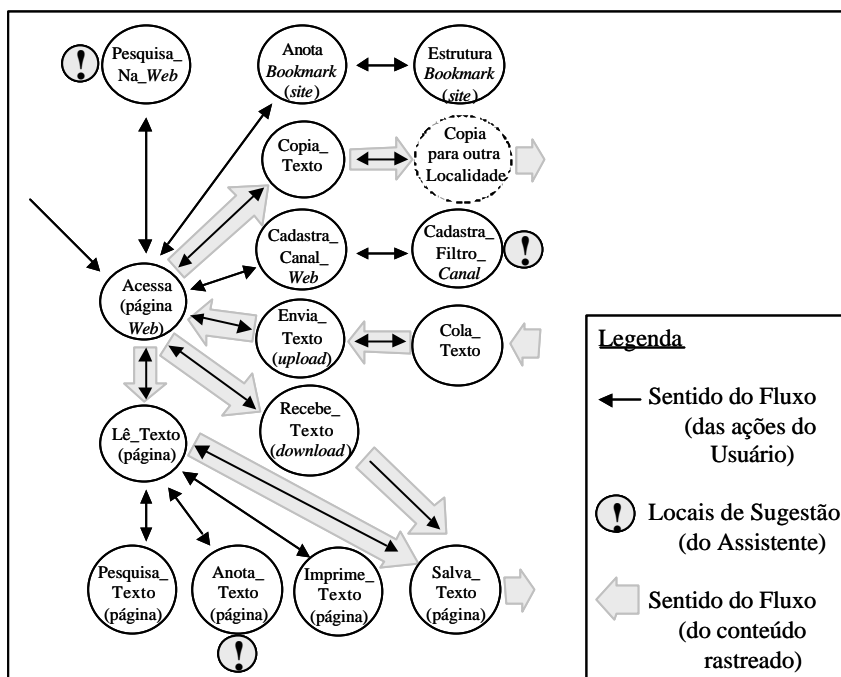


Figura 4.22: Exemplo de Modelo de Comportamento no Navegador Web

- Correio Eletrônico. A Figura 4.23 ilustra um exemplo de *Modelo de Mundo* do comportamento do usuário usando o Correio Eletrônico. Aqui, também se aplicam os comentários anteriores sobre o fluxo das ações percebidas, a atualização das probabilidades, e o fluxo do conteúdo rastreado. Nesta Localidade, o Assistente de *Feedback* pode fazer sugestão quando o usuário está compondo mensagem e/ou incluindo anexo na mensagem em composição.

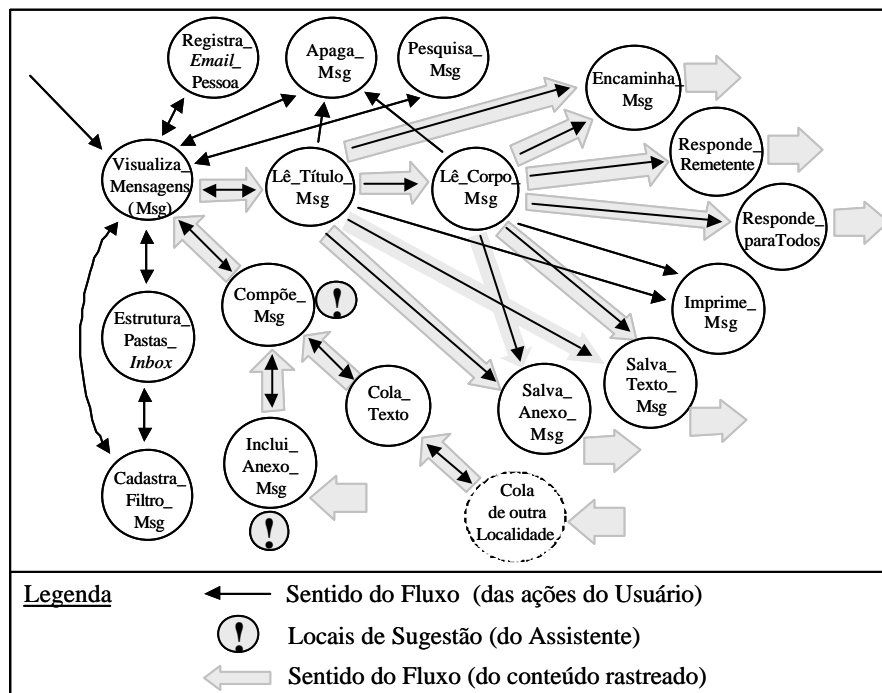


Figura 4.23: Exemplo de Modelo de Comportamento no Correio Eletrônico

- Gerenciador de Arquivos. Aqui o Assistente de *Feedback* não faz sugestão (para simplificar o cenário de modelagem deste estudo de caso). Assim, apenas monitora o fluxo de ações (atualizando probabilidades) e também o fluxo de conteúdo rastreado, tal como a Figura 4.24 ilustra um exemplo de *Modelo de Mundo* do comportamento do usuário usando o Gerenciador de Arquivos.

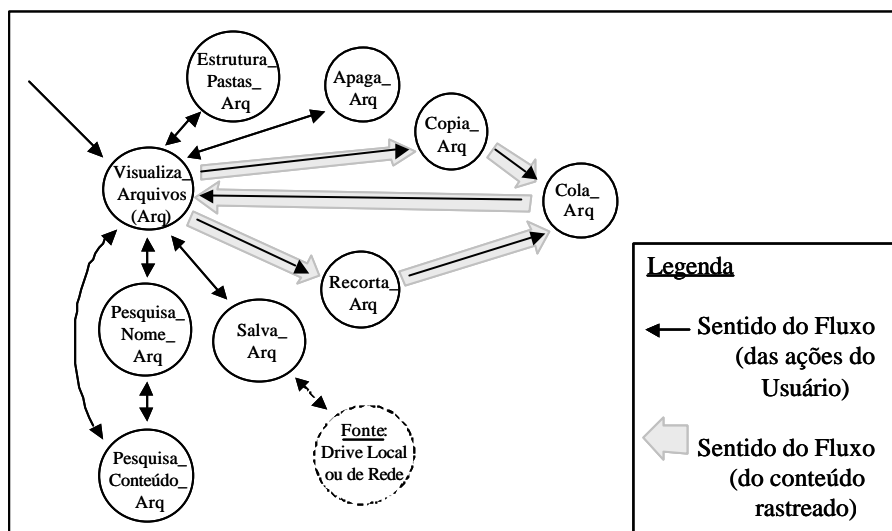


Figura 4.24: Exemplo de Modelo de Comportamento no Gerenciador de Arquivos

- Editor. Seguindo o padrão do projeto, aqui também são monitorados os fluxos de ações percebidas e o fluxo de conteúdo rastreado. Nesta Localidade, o Assistente de *Feedback* pode fazer sugestão quando o usuário está fazendo leitura/estudo e também quando o usuário usa a ferramenta de anotar texto (normalmente, o texto que está sendo estudado). Um exemplo de *Modelo de Mundo* do comportamento do usuário usando o Editor é ilustrado na Figura 4.25.

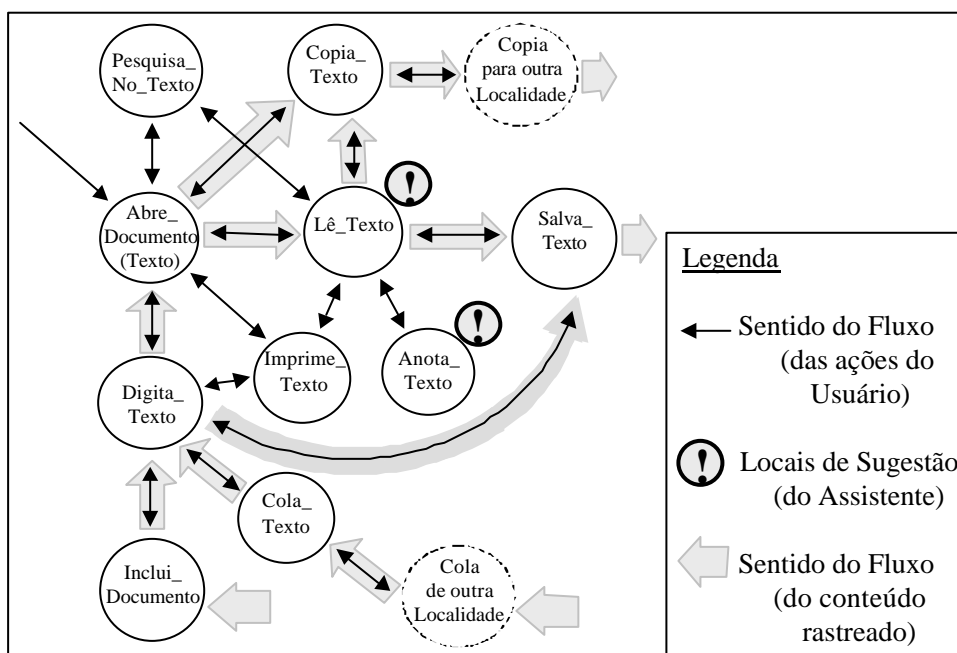


Figura 4.25: Exemplo de Modelo de Comportamento no Editor de Textos

- Agenda. Uma vez que todas as tarefas agendadas são evidentemente úteis e do interesse do usuário, esta Localidade é muito importante como provedora de

semântica, que é propagada para todas as ações (em outras Localidades) que se relacionam com o conteúdo das tarefas agendadas (conteúdo este que é rastreado). Assim como no Gerenciador de Arquivos, para simplificar este estudo de caso, o Assistente de *Feedback* não faz sugestão ao usuário. A Figura 4.26 ilustra um exemplo de *Modelo de Mundo* do comportamento do usuário usando a Agenda.

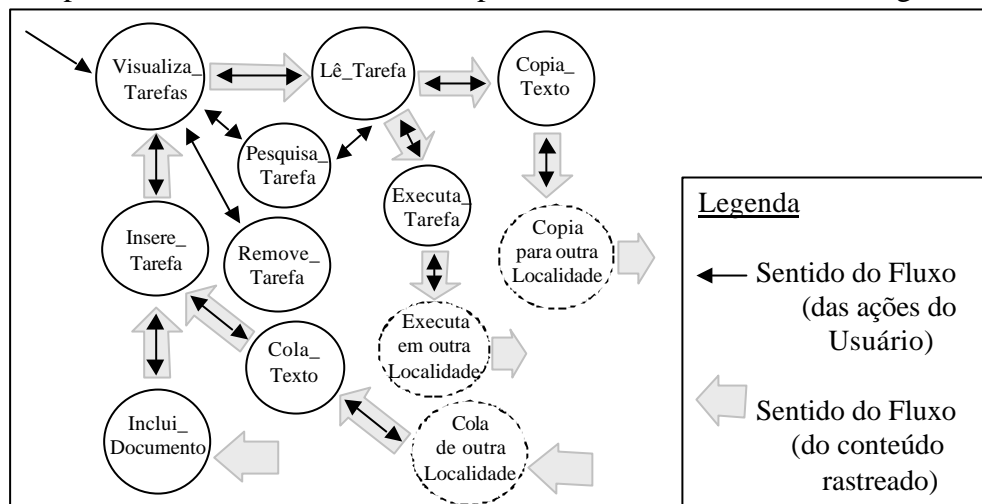


Figura 4.26: Exemplo de Modelo de Comportamento na Agenda

Até agora, foram apresentados exemplos de *Modelos de Mundo* que representam o comportamento do usuário nas Localidades do Ambiente do Assistente de *Feedback*. Note que estes modelos são dinâmicos e adaptativos (na Definição 2.3.1, Capítulo 3, foi descrito como os *Símbolos* mais antigos e não utilizados são “esquecidos”). Além disso, são modelos focalizados nas *Percepções das Ações (PA)* e que refletem o padrão de comportamento do usuário. Embora esteja focalizado nas *PA*, estas *PA* estão relacionadas (formam *Símbolos Compostos*) com as *Percepções dos Objetos (PO)* do Ambiente (neste caso, Objetos de Informação), conforme ilustra a Figura 4.18. Vale ressaltar que o rastreamento do conteúdo tem relação com a propagação de semântica no *Tempo Contextual*, sendo isto utilizado no *Serviço de Aprendizagem* (Seção 4.5). Ainda sobre a aprendizagem, é importante destacar que os *Símbolos Compostos (PO.PA)* das hierarquias de *PO* e de *PA* (a Figura 4.21 ilustra parte da *PA*) fazem uma generalização que é usada no *Serviço de Tomada de Decisão* do Assistente de *Feedback* (Seção 4.6).

Comportamento do Assistente de Feedback

Conforme enfatizado no início desta subseção, o Assistente de *Feedback* simula no *Modelo de Mundo* o comportamento do usuário para poder fazer suas sugestões – as quais correspondem às ações mais prováveis que o usuário faria no modelo simulado. Ou seja, o *Modelo de Mundo* do Assistente de *Feedback* inclui o *Modelo de Mundo* do usuário.

Neste momento é importante lembrar o papel do *Modelo de Mundo* no *Serviço de Detecção de Contexto*, mais especificamente no casamento da sintaxe e da semântica dos *Estímulos Externos* (ou *Percepções Incertas*). A Figura 4.27 ilustra o papel do *Modelo de Mundo*, onde num dado instante existe a entrada simultânea de *Percepção Certa* (p.ex. o usuário apagando um *email*) e de *Percepção Incerta* (p.ex. a relevância do conteúdo do *email* apagado). Além da entrada das *Percepções Certa* e *Incerta*, o *Modelo de Mundo* está fornecendo a semântica do estado de ação ativo (neste exemplo,

o Observa_Correio_Eletrônico, vide Figura 4.30). Portanto, no casamento entre sintaxe e semântica, o conteúdo do *email* apagado é rastreado a partir do estado de ação ativo (aqui, Observa_Correio_Eletrônico) e, neste exemplo, este mesmo conteúdo tinha sido transferido para um agendamento de compromissos (fato registrado numa entrada de percepção num T_C anterior). Assim, a *Percepção Incerta* (sobre a relevância do conteúdo do *email*) a princípio teria uma semântica indicando desinteresse (p.ex. o usuário apagou o *email*), todavia após o casamento da sintaxe e semântica, esta *Percepção Incerta* foi realimentada com uma semântica indicando interesse - porque o conteúdo deste *email* recém-apagado foi usado no agendamento de um compromisso (num instante T_C anterior).

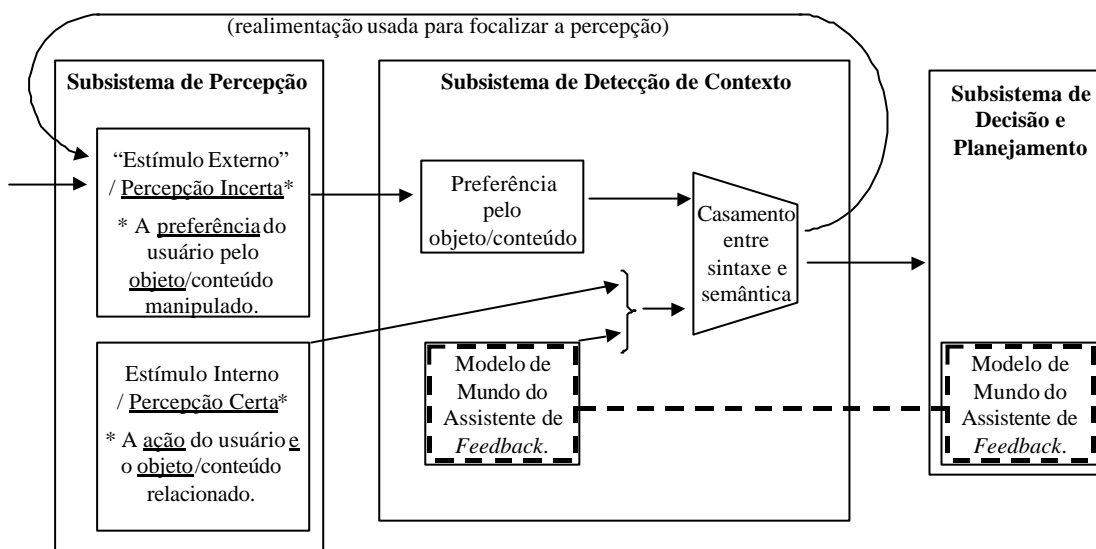


Figura 4.27: Processamento de Percepções Incertas no Assistente de *Feedback*

Apesar de aparecer em dois lugares diferentes na Figura 4.27, o *Modelo de Mundo* é único, sendo compartilhado pelo *Serviço de Detecção de Contexto* e também pelo *Serviço de Tomada de Decisão* – os quais exercem uma influência mútua (Teorema 2.2.2, Cap.3). O estado ativo do *Modelo de Mundo* influencia a semântica da *Percepção Incerta*, mas uma vez reconhecida a percepção, ela influencia na transição do estado ativo do *Modelo de Mundo* (isto é, influencia nas decisões). Veremos a seguir que, no *Ambiente* do Assistente de *Feedback*, o estado que representa a importância da ação do usuário (importância considerando o *workflow* do usuário) influencia na semântica associada à *Percepção Incerta* sobre a preferência do usuário (pelo objeto/conteúdo que está exercendo uma ação).

Inicialmente, abordando o 1^o nível do *Modelo de Mundo* do Assistente de *Feedback*, temos um estado que representa a operação contínua do Assistente – seja tratando as percepções de ações ‘?’ do usuário (‘Ação_Usuário:?’)³, ou a chegada de mensagens de *email* com conteúdo ‘?’ (‘Email:?’). A operação do Assistente é contínua mesmo quando a percepção de estímulos é vazia (\emptyset), porque existem ações *default* a serem executadas. A Figura 4.28 ilustra este 1^o nível do modelo.

³ Estamos seguindo a notação de Símbolo-Disparador ‘Símbolo:?’ (em Definição 2.2.2.4, Cap.3).

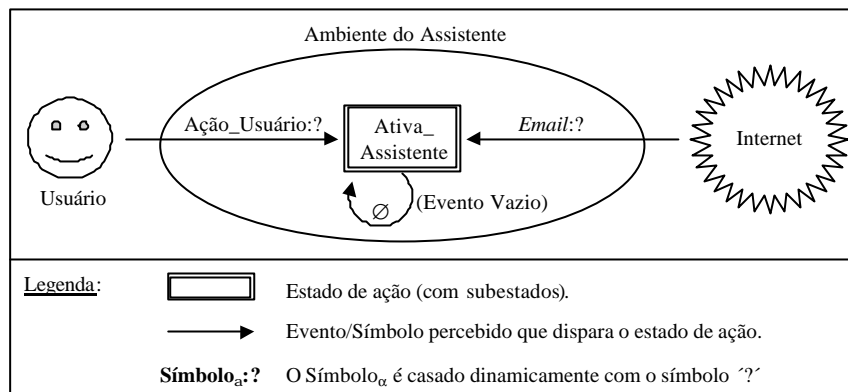


Figura 4.28: 1.º Nível do Modelo de Mundo do Assistente de *Feedback*

Nos níveis seguintes do *Modelo de Mundo*, se for percebido um evento *'Email:?'* ou um \emptyset (Evento Vazio) então o Assistente de *Feedback* ativa o estado *'Ações_Default'* onde são executadas ações de inferências, ações de refinamento de planos e ações de suporte (p.ex. busca de conteúdo novo para sugerir, etc). Todavia, se for percebido um evento *'Ação_Usuário:?'* então é ativado o estado *'Monitora_Ação'*. Neste estado, se for percebido que o local da ação do usuário é um local de sugestão *'Local_Sugestão:?'* (são os locais marcados com \odot nas Figuras 4.22, 4.23 e 4.25), então é ativado o estado *'Faz_Sugestão'*. Ainda no estado *'Monitora_Ação'*, se for percebido que o local da ação do usuário é um outro local qualquer, então é ativado o estado *'Modela_Comportamento'*. A Figura 4.29 ilustra estes níveis seguintes do *Modelo de Mundo* do Assistente. Note que para definir a semântica da *Percepção Incerta*, o processo de focalização de percepção (por exemplo, percebe a ação do usuário, depois percebe o local da ação, etc) envolve uma realimentação representada pela seta curva na Figura 4.27. Revendo o exemplo usado anteriormente (sobre a ação do usuário apagar um *email*), a primeira percepção é "Ação_Usuário:Remove_Msg" que ativa o estado "Monitora_Ação" (que é um subestado de "Ativa_Assistente", Figura 4.29). Monitora_Ação, por sua vez, possui subestados (Figura 4.29, parte inferior), sendo que a segunda percepção é "Outro_Local:Correio_Eletrônico" (que ativa o subestado Modela_Comportamento).

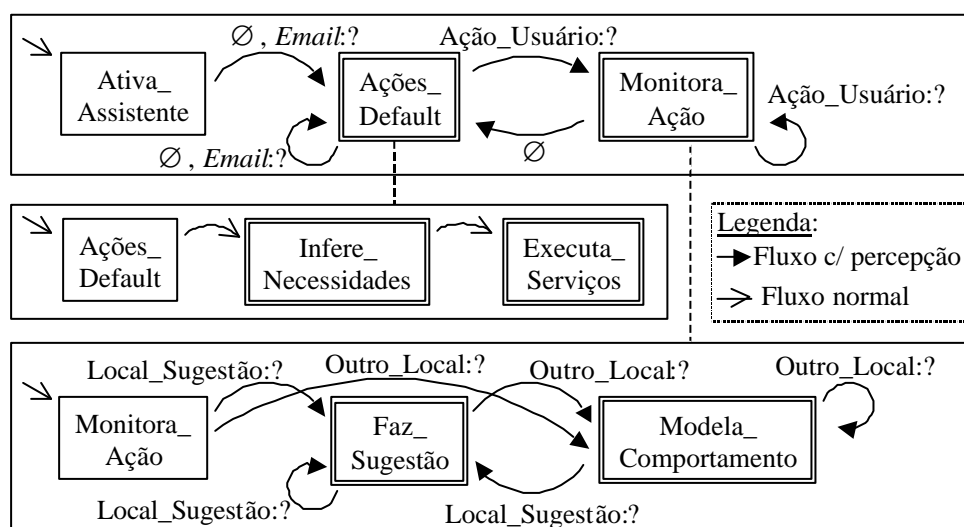


Figura 4.29: Níveis Seguintes do Modelo de Mundo do Assistente de *Feedback*

4.4.3 Relacionamento entre Percepção e Modelo de Mundo

Seguindo o objetivo de ilustrar o uso do *Serviço de Detecção de Contexto* SADI (Definição 2.2.2, Cap.3) neste estudo de caso, vamos focalizar a descrição do *Modelo de Mundo* do Assistente de *Feedback* considerando o estado 'Modela_Comportamento'. É interessante observar que os *Estímulos Externos* são *Percepções Incertas* (Definição 2.2.1.8, Cap.3). Todavia, nem toda *Percepção Incerta* é um *Estímulo Externo*, tal como mostra o cenário deste estudo de caso, porque nesta versão do Assistente de *Feedback* todos os eventos são *Estímulos Internos* que possuem *Percepção Certa* (não existe dúvida sobre quais as ações do usuário e qual o conteúdo de informação relacionado com estas ações). Aqui neste estudo de caso, a *Percepção Incerta* significa a percepção do grau de preferência/interesse do usuário (pelo objeto/conteúdo que está exercendo uma ação). Portanto o *Serviço de Detecção de Contexto* usa os subestados contidos no estado 'Modela_Comportamento' para associar semântica a esta *Percepção Incerta*.

O estado de ação 'Modela_Comportamento', ao ser ativado pelo *Símbolo-Disparador* 'Outro_Local:?' (Figura 4.29), recebe o *Símbolo* '?' já instanciado em algum estado dos modelos mostrados da Figura 4.22 até a Figura 4.26 (exceto os estados que representam locais de sugestão do Assistente ①). De acordo com a Localidade do *Símbolo* instanciado '?', um subestado de 'Modela_Comportamento' é ativado (em algum destes: *Observa_Correio_Eletrônico*, *Observa_Navegador*, *Observa_Editor*, *Observa_Agenda*, ou *Observa_Gerenciador_Arquivos*) - ou seja, a percepção influencia a transição de estado no *Modelo de Mundo*. Por outro lado, conforme mostra a Figura 4.30, dependendo do estado ativo (p.ex. 'Observa_Editor'), da ação percebida (p.ex. 'Salvou:?' e do destino da ação (p.ex. na Pasta_Relatório), é definida a semântica do grau de interesse do usuário pelo conteúdo '?' relacionado com a ação percebida (neste exemplo, é atribuído o grau de interesse '+10' ao conteúdo do Documento salvo na Pasta_Relatório).

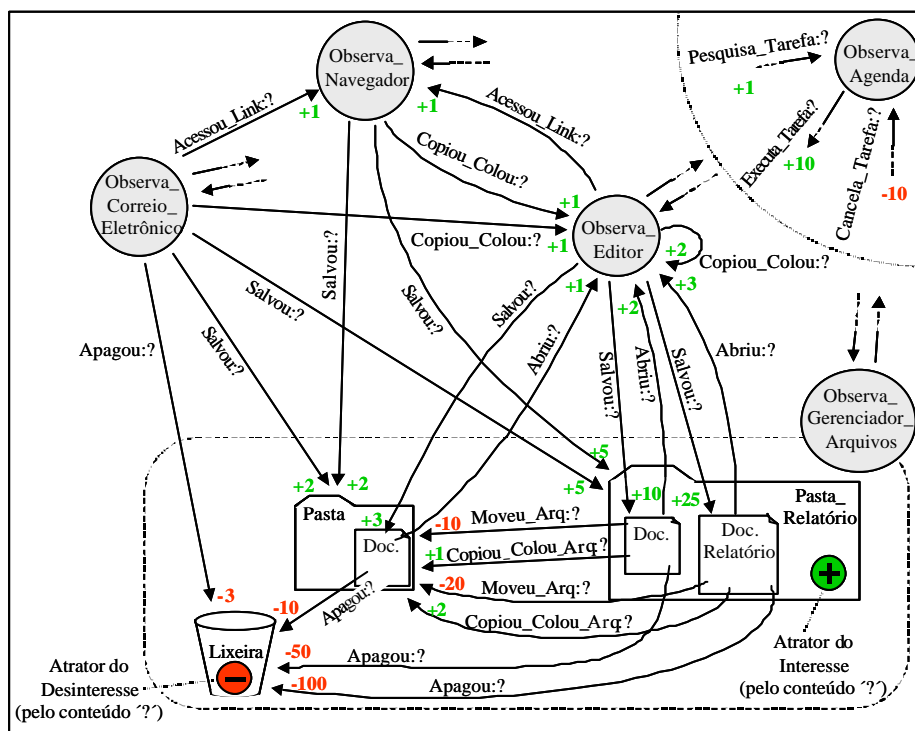


Figura 4.30: Subestados de ação embutidos no estado 'Modela_Comportamento'

A Figura 4.30 mostra como o *Modelo de Mundo* define a semântica do fluxo do conteúdo rastreado (ilustrado pelas setas largas, da Figura 4.22 até a 4.26). A semântica deste fluxo considera o estado de origem, a percepção da ação sobre um conteúdo ‘?’ (‘Ação_Percebida:?’), e o estado de destino. Note que ‘Ação_Percebida’ corresponde ao conceito de *Relacionamento-Transição* (em Definição 2.2.2.4, Cap.3), no sentido que faz a ligação entre o *Modelo de Mundo* e o *Espaço de Percepção*, atribuindo dinamicamente a semântica ao conteúdo ‘?’. Cabe diferenciar que, neste estudo de caso, o *Relacionamento-Transição* atribui semântica a uma *Percepção Incerta* (sobre a preferência do usuário) de um *Estímulo Interno* (a ação do usuário sobre um objeto de informação).

Numa eventual versão futura, o Assistente de *Feedback* pode conter *Relacionamentos-Transição* para atribuir semântica às *Percepções Incertas* de *Estímulos Externos*, tal como imagens da expressão facial do usuário e também comandos/expressões de voz do usuário. Neste caso, pode ocorrer também do Assistente usar *Relacionamentos-Transição* para atribuir semântica a uma *Percepção Incerta* (sobre a preferência do usuário) interpretada a partir de um *Estímulo Externo* (imagem ou som, que o Assistente também vai usar *Relacionamentos-Transição* para atribuir semântica no processo de reconhecimento do *Símbolo*). Trata-se de 2 ordens de *Relacionamentos-Transição*: os de 1ª ordem lidam com a incerteza de *Estímulos Externos*; já os de 2ª ordem lidam com a incerteza da interpretação de *Símbolos* reconhecidos (incluindo *Estímulos Internos* - que são *Percepções Certas*, como neste estudo de caso: o Assistente não tem dúvida sobre as ações do usuário e os objetos das ações, mas tem dúvida sobre a preferência do usuário). Note que os de 1ª ordem estão associados com o reconhecimento de *Símbolos* e os de 2ª ordem estão associados com a interpretação subjetiva dos *Símbolos* reconhecidos.

Ainda na Figura 4.30, o estado *Observa_Agenda* está ligado com todos os demais (para simplificar a figura, as ligações estão representadas pelas setas parcialmente tracejadas ---->). Na análise do *Ambiente* do Assistente de *Feedback*, as tarefas agendas são obviamente importantes para o usuário. Desta forma, quando o usuário acessa a agenda e a partir dela inicia uma tarefa agendada, então o conteúdo relacionado com esta tarefa recebe o valor ‘+10’ (definido aqui de forma empírica, através da análise conceitual do domínio). A plataforma de software pode dar suporte ao uso de *links* nas tarefas agendadas, que uma vez ativados, abrem a aplicação carregando automaticamente o conteúdo a ser trabalhado). Por outro lado, se uma tarefa é cancelada, então seu conteúdo é associado a um grau de desinteresse (valor ‘-10’). Da mesma forma realizada para os eventos percebidos (Figuras 4.12 e 4.13), os valores definidos aqui para a semântica (que mede o grau de interesse do usuário pelo conteúdo sendo manipulado) são baseados no conhecimento de especialistas humanos, o mesmo critério usado em [HEC 98], pois em ambos os casos os *Contextos* do domínio da aplicação estão bem delimitados. Mais adiante, na Seção 4.6, é descrito como estes valores são usados para fazer as sugestões ao usuário.

Portanto, na medida em que o usuário interage com o *Ambiente* do Assistente de *Feedback*, são construídos os relacionamentos entre *Percepções de Ações (PA)* do usuário, *Percepções de Objetos (PO)* de informação manipulados pelo usuário, e *Percepções do Feedback Interior (PI)* do quanto o usuário está provavelmente interessado/desinteressado pelo conteúdo objeto da ação. Estes relacionamentos (PO,PA,PI) formam os cubos básicos do espaço multidimensional ilustrado nas Figuras 4.18 e 4.19.

4.5 Serviço de Aprendizagem do Agente de Software

Conforme exposto na Definição 2.2.4 (Capítulo 3), o processo de aprendizagem do SADI ocorre com a ativação/detecção de contextos, onde no *Tempo Contextual* T_C ocorre o *Fluxo de Feedforward* ($PO_{T_C}.PA_{T_C}$), e depois no *Tempo Contextual* T_{C+1} ocorre o *Fluxo de Feedback* ($(PO_{T_C}.PA_{T_C}).PI_{T_{C+1}}$). Cabe lembrar também que a relação de causa e efeito no *Tempo Contextual* é relativa a cada contexto ativado e que a *Entrada de Percepção* (PO, PA e PI) no SADI tem um *Uso Duplo* (Definição 2.2.1.4, Cap.3), o que significa que a *Entrada de Percepção* simultaneamente ativa um contexto e também carrega consigo a *Percepção de Feedback Interior* (PI) de possíveis efeitos/resultados das ações realizadas neste contexto ativado.

Nas seções anteriores, vimos como o *Serviço de Percepção* e o *Serviço de Detecção de Contexto* constroem as relações ($(PO_{T_C}.PA_{T_C}).PI_{T_{C+1}}$) para o cenário do Assistente de *Feedback* deste estudo caso. Embora tenha sido relacionada no cubo básico (PO, PA, PI) e ilustrada nas Figuras 4.18 e 4.19, a *Percepção de Feedback Interior* (PI) foi parcialmente descrita na seção anterior porque foi necessário aguardar a descrição do *Modelo de Mundo* do Assistente de *Feedback*. Isto ocorreu porque a PI possui uma relação semântica muito estreita com o estado de ação ativo do *Modelo de Mundo*.

Portanto, é importante ressaltar que em cada cenário exposto no *Modelo de Mundo* do Assistente de *Feedback* (na Subseção 4.4.2), o *Serviço de Aprendizagem* estava operando integrado com o *Serviço de Detecção de Contexto*, ao participar de cada ciclo de operação (\rightarrow Percepção \rightarrow Detecção de Contexto \rightarrow Aprendizagem \rightarrow Decisão). Agora, nesta seção, vamos comentar algo mais sobre o rastreamento de conteúdo (na Subseção 4.5.1) e depois descrever como a PI é mapeada na relação de causa e efeito no *Tempo Contextual* ($(PO_{T_C}.PA_{T_C}).PI_{T_{C+1}}$) do cenário deste estudo de caso (na Subseção 4.5.2).

4.5.1 Rastreamento do Conteúdo

Na Definição 2.3 (Capítulo 3) foi descrito que, além dos *Relacionamentos Semânticos Multidimensionais*, o modelo de *Representação de Conhecimento* do SADI possui simultaneamente os *Relacionamentos Causais* (isto é, as relações de dependência condicional) e as *Estruturas* (de índices) de *Acesso aos Dados*. Isto torna o SADI conceitualmente similar a (o que poderia ser referido como) um *Datawarehouse* Orientado a Objetos que opera de forma contínua e incremental usando os Serviços SAIC⁴. Esta similaridade conceitual se refere a: 1) Dados Orientados a Contexto; 2) Vários dados sumarizados e sintetizados; 3) Várias estruturas de índices; 4) Representação de conhecimento multidimensional; e 5) Armazenamento relativamente grande – proporcional ao escopo do domínio.

Aqui no cenário do estudo de caso, para alcançar o objetivo do Assistente de *Feedback* (Seção 4.1), existe outra estrutura de índices no espaço multidimensional, além da indexação dos cubos básicos pelas hierarquias de *Percepções de Ações* (PA) e de *Percepções de Objetos* (PO). Esta outra indexação adicional é “transversal” a todas as outras indexações, no sentido que sua organização hierárquica é feita na perspectiva do uso do conteúdo rastreado (Conteúdo Filtrado e Conteúdo não-Filtrado, dos Contextos de Sugestão descritos na Seção 4.1). Portanto, esta indexação possui como ponto de entrada as PO básicas deste cenário, que são os Termos (palavras-chave). Ou

⁴ Assim como [ABO 99], que constrói de forma incremental histogramas que se auto-ajustam (*self-tuning histograms*) e são usados para otimização de consultas de banco de dados.

seja, ao acessar um Termo T nesta hierarquia de índices, sua primeira entrada generaliza todos as *Percepções de Feedback Interior (PI)* tanto para o *Feedback* da classe Bom/Certo/Sucesso (aqui significa o interesse do usuário por T) quanto para o *Feedback* da classe Ruim/Errado/Insucesso (aqui significa o desinteresse do usuário por T). Mas esta primeira entrada possui ponteiros para os contextos onde T ocorreu, sendo que cada ocorrência de T generaliza os *Feedbacks* dos níveis abaixo e também aponta para estas ocorrências. A seguir veremos como esta indexação na perspectiva do conteúdo é usada na aprendizagem (e também na tomada de decisão, na Seção 4.6).

4.5.2 *Feedback Interior Indireto e Direto*

Ao descrever como a *Percepção de Feedback Interior PI* é mapeada na relação de causa e efeito no *Tempo Contextual* $((PO_{Tc}.PA_{Tc}).PI_{Tc+1})$ do cenário deste estudo de caso, é preciso considerar que a *PI* está associada a dois universos de ações:

- As Ações do usuário relacionadas com os Objetos do Ambiente deste usuário. Neste universo, a *Percepção do Feedback Interior* pelo Assistente é indireta (**PI Indireta**), ou seja, o Assistente de *Feedback* coleta indicadores de prováveis preferências (+) ou não-preferências (-) para o conteúdo *PO* sendo manipulado, de acordo com os eventos percebidos (Figuras 4.12 e 4.13) e o rastreamento do conteúdo (Figura 4.30).
- As Ações de Sugestão do Assistente de *Feedback* relacionadas com ‘As Ações do usuário relacionadas com os Objetos do Ambiente deste usuário’. Neste universo, a *Percepção do Feedback Interior* pelo Assistente é direta (**PI Direta**), ou seja, o Assistente coleta respostas diretamente do usuário através dos *Feedbacks* de:
 - Curto prazo: através das janelas de *Feedback Rápido* (descritas na Seção 4.1). Aqui, novamente os valores de indicadores são definidos de forma empírica através de uma análise conceitual do domínio. Quando o usuário aceita uma sugestão, a *PI Direta* (para o conteúdo *PO* sendo manipulado) provê um indicador de preferência (+10), e acrescenta no modelo de comportamento do usuário o padrão de ações percebidas *PA* relacionadas com o conteúdo *PO* da sugestão. Caso contrário, se o usuário rejeita uma sugestão, então o valor do indicador de preferência é (-10) e o padrão de ações *PA* relacionadas é subtraído do modelo de comportamento.
 - Prazo maior: através da interface de *Feedback* do Serviço de Canais Filtrados. A sugestão que o Assistente de *Feedback* faz aqui se baseia na *PI Indireta* (coletada continuamente) e na *PI Direta* (coletada nas janelas de *Feedback* rápido). Aqui a *PI Direta* é a mesma do caso anterior (o *Feedback* de curto prazo), exceto para os valores de preferência: (+25) quando o usuário aceita a sugestão e (-25) quando ele rejeita.

Existe uma relação de influência mútua entre os padrões de comportamento do usuário e as *PI Indireta* e *PI Direta*. Na *PI Indireta*, os padrões de comportamento das *PA* fornecem indicadores sobre as preferências das *PO*. Mas na *PI Direta*, a confirmação das preferências das *PO* fornece indicadores sobre quais padrões de comportamento das *PA* estão relacionados com as novas preferências das *PO*. Trata-se de um mecanismo de realimentação contínua da aprendizagem dos padrões de comportamento que refletem as preferências do usuário.

Durante a coleta de *PI Indireta*, cada Termo T da indexação do conteúdo rastreado vai apontar para onde T ocorreu e vai receber o *PI Indireto* – o qual é

generalizado na indexação de T . Ou seja, cada T possui um mapeamento do padrão de comportamento do usuário.

Após um período P de acúmulo de $PI_Indireta$ fornecida pelos padrões de comportamento do usuário, o Assistente faz sugestões que recebem PI_Direta através das janelas de *Feedback Rápido* (e também quando o usuário ativar a interface de *Feedback* do Serviço de Canais Filtrados). Para cada Termo T sugerido pelo Assistente ao usuário (nos locais de sugestão mostrados na seção anterior), a PI_Direta vai revelar a preferência real do usuário por T e também vai fornecer indicações sobre quais os padrões de comportamento que refletem a preferência do usuário (uma vez que T aponta para um conjunto de ações relacionadas).

Para cada Termo T rastreado, ao se considerar a dimensão de *Tempo Contextual* na perspectiva da aprendizagem dos padrões de comportamento (através do PI_Direto), no *Tempo Contextual* T_C várias ações são realizadas (e cada uma recebendo $PI_Indireto$) sobre o mesmo conteúdo T que está sendo rastreado (ou seja, em T_C existe um padrão de comportamento). Mas quando o Assistente sugere T , o evento do usuário prover o PI_Direto significa o *Feedback* no *Tempo Contextual* T_{C+1} para o padrão de comportamento ocorrido em T_C . A Figura 4.31 ilustra o rastreamento neste processo de aprendizagem, considerando a seguinte seqüência de ações do usuário sobre o Termo T rastreado que está contido num conteúdo C :

→ Navegador.Acessa_Página(T) → Navegador.Copia_Texto(T) → Editor.Cola_Texto(T)
 → Editor.Digita_Texto(T) → Editor.Anota_Texto(T) → Editor.Aceita.AF_Sugere_Anotação(T)

Esta seqüência significa que o usuário acessou uma página web (e salvou seu conteúdo), leu esta página e copiou um trecho para o editor, onde redigiu um documento e fez anotações neste documento (inclusive aceitando a sugestão do Assistente de incluir o Termo T em uma das anotações que fez).

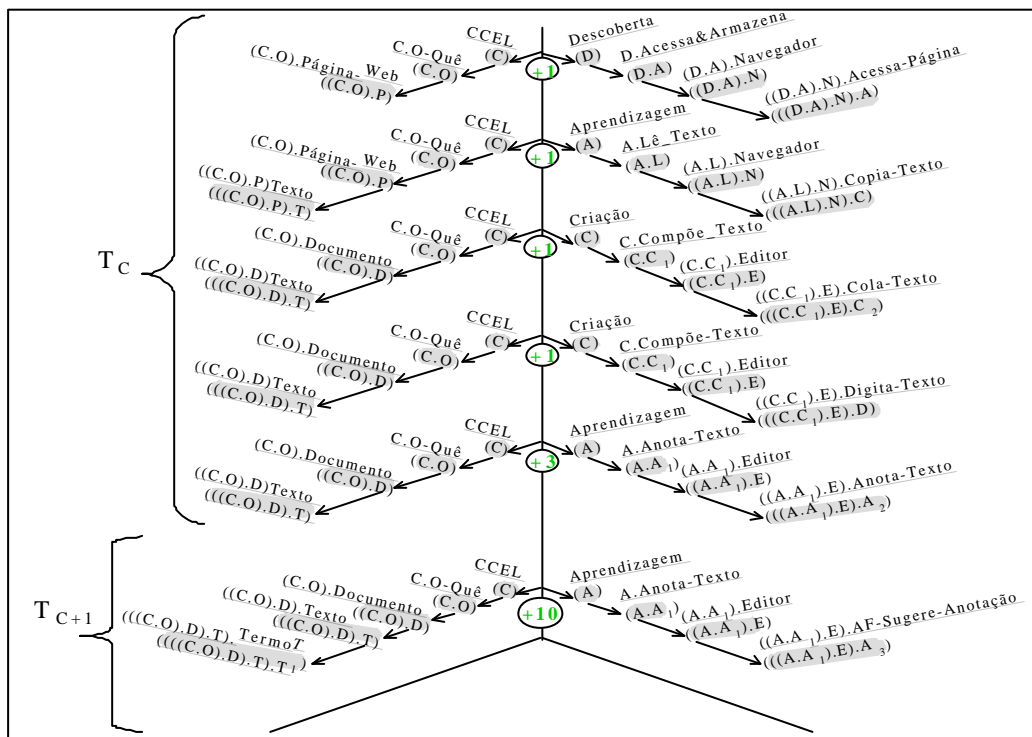


Figura 4.31: Rastreamento de Padrão de Comportamento Relacionado com o Termo T

Na Figura 4.31, a *Percepção das Ações (PA)* segue a hierarquia de classes do Contexto de Funcionalidades (Figura 4.8). Mas para simplificar a visualização desta figura, com relação à *Percepção dos Objetos (PO)*, somente a subclasse O_Quê da hierarquia de Classes CCEL (Figura 4.15) é mostrada. Ainda para simplificar a visualização, as ligações dos relacionamentos ((PO.PA).PI) são omitidas, apenas os valores \oplus de *PI* são mostrados (*PI_Indiretas* em T_C e *PI_Direta* em T_{C+1}).

Ainda na Figura 4.31, quando no *Tempo Contextual* T_{C+1} o usuário aceitou a sugestão do Termo T que o Assistente fez, o usuário então proveu simultaneamente a *PI_Direta* para o Termo T e também para o padrão de comportamento ocorrido no *Tempo Contextual* anterior T_C . A Figura 4.32 mostra a respectiva indexação do Termo T que incorpora a *PI_Indireta* (indicadores de preferência) e a *PI_Direta* (indicadores de preferência e padrão de comportamento) ocorridas nos *Tempos Contextual* T_C e T_{C+1} .

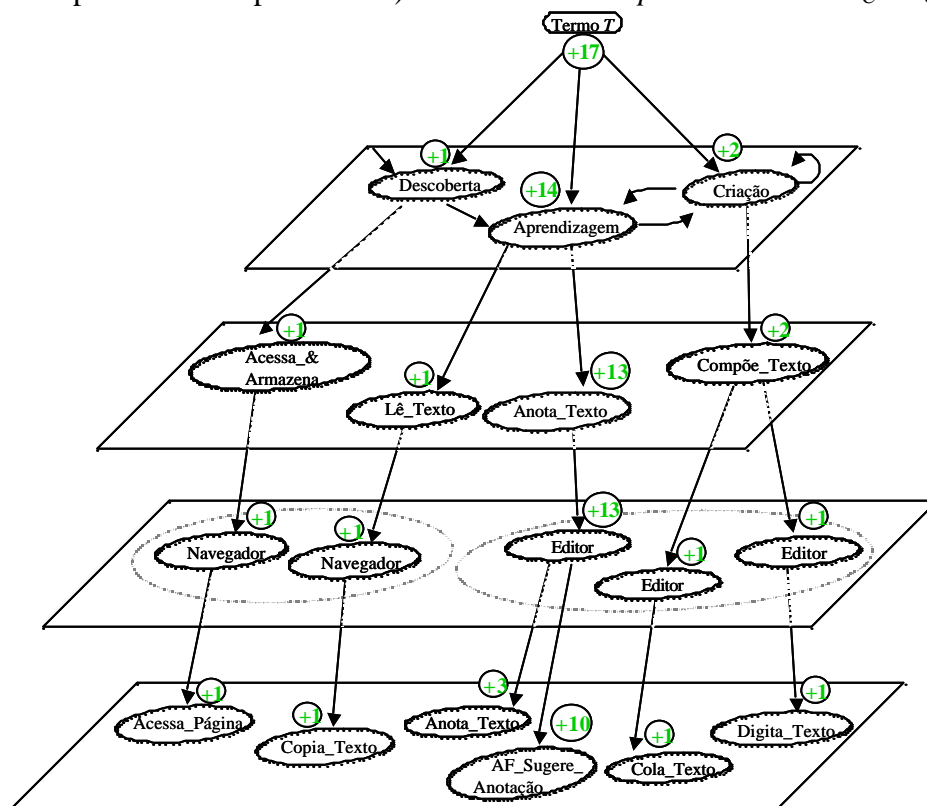


Figura 4.32: Indexação do Termo T rastreado nos *Tempos Contextual* T_C e T_{C+1}

Este processo de aprendizagem ocorre em cada localidade descrita no *Modelo de Mundo* do Assistente de *Feedback* (Subseção 4.4.2), refletindo o comportamento do usuário (conforme ilustrado nas Figuras 4.22 – 4.26). Este comportamento é atualizado de forma contínua e incremental através da relação de influência mútua entre a *PI_Indireta* e a *PI_Direta* – onde, por um lado, os padrões de comportamento das *PA* fornecem indicadores sobre as preferências das *PO*; e, por outro lado, a confirmação das preferências das *PO* sugeridas fornece indicadores sobre quais padrões de comportamento das *PA* estão relacionados com as novas preferências das *PO*.

Esta organização contextual suporta a *Aprendizagem por Reforço* do Assistente de *Feedback*, minimizando o Problema da Lacuna (Seção 2.8) através do uso de controle semântico no *Espaço de Contexto* e das relações de causa-e-efeito entre os *Fluxos de Feedforward* e de *Feedback* na dimensão de *Tempo Contextual*.

4.6 Serviço de Tomada de Decisão do Agente de SW

Ao tratar das decisões e planejamentos do Assistente de *Feedback* é preciso considerar o objetivo principal deste (descrito na Seção 4.1), que é sugerir ao usuário dos Canais Filtrados:

- A remoção de Termos provavelmente inúteis (que já estão nos Filtros). Este é Contexto_de_Exclusão, onde são considerados os Termos do Conteúdo Filtrado;
- A inclusão de Termos provavelmente úteis (que ainda não estão nos Filtros). Este é o Contexto_de_Inclusão, onde são considerados os Termos do Conteúdo não-Filtrado;

Esta sugestão ocorre quando o usuário ativar a interface de *feedback* do Serviço de Canais Filtrados. Portanto, para cada Termo T rastreado em cada *Contexto* (Exclusão/Inclusão), o Assistente de *Feedback* consulta a estrutura de índices (descrita na subseção 4.5.1, e ilustrada em parte, na Figura 4.32) para obter os valores dos indicadores de interesse (+) e desinteresse (-). Assim, no Contexto_de_Exclusão, o Assistente de *Feedback* sugere n Termos T com os maiores indicadores de desinteresse (-) ou eventualmente, aqueles T com menores indicadores de interesse. Quanto ao Contexto_de_Inclusão, são sugeridos n Termos T com os maiores indicadores de interesse (+) ou eventualmente, aqueles T com menores indicadores de desinteresse.

Este processo de sugestão é simples e determinístico: quando o usuário ativar a interface, o Assistente de *Feedback* consulta a indexação de Termos e seleciona n para cada *Contexto*, seguindo uma ordem de valores de indicadores.

Todavia, conforme exposto ao longo deste capítulo, o processo de indexação do conteúdo rastreado envolve um sistema complexo, onde existe uma relação de influência mútua entre os padrões percebidos de comportamento do usuário (PA , relacionados com PO) e as *Percepções de Feedback Interior* (PI , que pode ser *Indireta* e *Direta*, Subseção 4.5.2), dentro de *Contextos* detectados, seguindo uma relação de causa e efeito numa dimensão de *Tempo Contextual* T_C .

Além disso, o processo de sugestão na interface de *feedback* do Serviço de Canais Filtrados é subsidiado pelas sugestões nas janelas de *Feedback Rápido* (Seção 4.1) e envolve **Recursos de Ações**, isto é, são aqueles necessários para realizar uma ação. No escopo do Assistente de *Feedback*, estes Recursos variam para cada tipo de ação:

- Ação de Percepção (das Ações do Usuário; e das suas Próprias Ações como Assistente): conforme descrito na Seção 4.3, os recursos necessários para perceber estes eventos são os *parsers* e códigos de métodos, funções e macros embutidos nas ferramentas/aplicações. Estes recursos formam o **Comportamento Reativo** do Assistente de *Feedback*.
- Ação de Modelagem (do comportamento/preferências do usuário): conforme descrito nas Seções 4.4 e 4.5, os recursos necessários para modelagem são o *Serviço de Detecção de Contextos* e o *Serviço de Aprendizagem*. Cabe lembrar que o *Modelo de Mundo* do Assistente também é utilizado nesta Ação de Modelagem. Estes recursos também participam do *Comportamento Reativo* do Assistente.
- Ação de Sugestão (de *Feedback Rápido*): ocorre no Contexto de Localidades (Navegador, Correio, Editor, Agenda e Gerenciador de Arquivos), nos Locais de

Sugestão (!) mostrados nas Figuras 4.22 - 4.26 (excetuando a interface de Canais Filtrados). Os recursos necessários para a Ação de Sugestão são os Conteúdos Candidatos (descritos a seguir, na Subseção 4.6.1) obtidos através de um processo que envolve ações de decisão e de Planejamento. Estes recursos são usados no comportamento de *Sistema Cognitivo* no Assistente (segundo o conceito de [CRU 2003], pela habilidade de planejar adiante usando um *Modelo de Mundo*). Todavia, a Ação de Sugestão também participa do *Comportamento Reativo*, uma vez que ela é uma reação a um *Estímulo Interno*. Ou seja, a percepção de uma determinada ação do usuário pode disparar o processo de planejar uma forma de encontrar o que este usuário provavelmente necessita.

4.6.1 Conteúdo Candidato

O **Conteúdo Candidato (Conteúdo_C)** é constituído por Objetos de Informação⁵ e serve de base para a Ação de Sugestão do Assistente de *Feedback*, oferecendo as opções de escolha para o Assistente decidir o que vai sugerir ao usuário. O Conteúdo_C pode ser obtido no *Ambiente Interno* (Conteúdo_{C-I}) ou no *Ambiente Externo* (Conteúdo_{C-E}), dependendo dos *Contextos* de Funcionalidades e Localidade onde está o Local de Sugestão:

- Localidade: Navegador
 - Funcionalidade: Aprendizagem
 - Sugestão: de Termos para anotar (ação: AF_Sugere_Anotação);
 - Conteúdo_{C-I}: são os Termos contidos na página Web sendo lida e, se existir, Termos Propagados (de outras ações recentes do usuário) e que estão também contidos na página em questão;
 - Funcionalidade: Seleção
 - Sugestão: de Termos para pesquisar nas máquinas de busca da *Web* (ação: AF_Sugere_Pesquisa);
 - Conteúdo_{C-I}: são os Termos Propagados de outras ações recentes do usuário.
- Localidade: Correio Eletrônico
 - Funcionalidade: Comunicação/Interação com Pessoas
 - Sugestão: de inclusão de endereço de *email* (em mensagem sendo enviada – podendo ser uma mensagem nova, ou um *forward*), ação: AF_Sugere_email_Pessoa;
 - Conteúdo_{C-I}: são os endereços de *email* que o usuário possui um histórico de interações.
 - Funcionalidade: Comunicação/Interação com Pessoas
 - Sugestão: de inclusão de documento em anexo de *email* (em mensagem sendo enviada – podendo ser uma mensagem nova, ou um *forward*), ação: AF_Sugere_Anexo;
 - Conteúdo_{C-I} e Conteúdo_{C-E}: C-I) documentos armazenados/salvos pelo usuário no sistema de arquivos local; e, C-E) documentos descobertos pelo Assistente de *Feedback* na *Web*.
- Localidade: Editor

⁵ Por exemplo: notícias, *emails*, páginas *Web*, documentos, palavras-chave/Termos, etc.

- Funcionalidade: Aprendizagem
 - Sugestão: de Termos para anotar (ação: AF_Sugere_Anotação);
 - Conteúdo_{C-I}: são os Termos contidos no documento sendo lido e, se existir, Termos Propagados (de outras ações recentes do usuário) e que estão também contidos no documento em questão;
- Funcionalidade: Aprendizagem

Sugestão: de documentos para serem lidos (ação: AF_Sugere_Leitura);

 - Conteúdo_{C-I} e Conteúdo_{C-E}: C-I) documentos armazenados/salvos pelo usuário no sistema de arquivos local; e, C-E) documentos descobertos pelo Assistente de *Feedback* na *Web*.

Quanto ao Conteúdo Candidato obtido no Ambiente Interno (Conteúdo_{C-I}), ele é distribuído/replicado através do serviço do *Serviço de Percepção do Ambiente do Agente de Software* (Seção 4.3). Este mesmo serviço de distribuição/replicação atende de uma só vez estas três demandas (formar *Símbolos Compostos*, propagar Conteúdo_{C-I} e manter Planejamentos atualizados). A Figura 4.11 mostra um exemplo de distribuição de Conteúdo_{C-I} no Contexto de Funcionalidades ‘Seleção/Pesquisa’.

De uma forma geral, a propagação de Conteúdo_{C-I} ocorre em: i) todos os Contextos de Funcionalidades (além do exemplificado: ‘Seleção/Pesquisa’); e ii) todos os Contextos de Localidades. Mas existe uma diferença fundamental entre o Conteúdo_{C-I} e o conteúdo (PO,PA,PI) no espaço de percepções:

- O conteúdo (PO,PA,PI) ocorre num Contexto detectado/ativado e faz parte da base de conhecimento “consciente” do Assistente, isto é, pode-se acessar uma instância qualquer (PO,PA,PI) através das estruturas de índices existentes. Além disso, o conteúdo (PO,PA,PI) fica armazenado quase que definitivamente (quase, porque existe um mecanismo de remoção de *Símbolos* muito antigos e pouquíssimos usados – considerando a marcação de tempo na dimensão de *Tempo Contextual T_C*).
- Já o Conteúdo_{C-I}, ele é originado de uma instância (PO,PA,PI) de um Contexto detectado/ativado, mas é distribuído/replicado ao mesmo tempo nas dimensões dos Contextos de Funcionalidades e Localidades. Todavia, mesmo sendo indexado da mesma forma que uma instância (PO,PA,PI), o Conteúdo_{C-I} faz parte do conhecimento “inconsciente” do Assistente (ou seja, só é acessado pela Ação de Sugestão do Contexto específico dos Locais de Sugestão) e além disso o Conteúdo_{C-I} tem uma duração de curto prazo (apenas uma contagem de *Tempo Contextual T_C*). Ou seja, cada Local de Sugestão em Contextos não-ativados vai acumulando as propagações de Conteúdo_{C-I} dos Contextos que são ativados e, no momento que o Contexto de um Local de Sugestão é ativado, então a Ação de Sugestão usa o Conteúdo_{C-I} recebido. Depois disso, o Conteúdo_{C-I} é descartado, e somente é incorporado no espaço de percepções (PO,PA,PI) se o Assistente usá-lo para fazer uma sugestão (a qual vai receber um *Feedback* do usuário).

No Ambiente do Assistente de *Feedback*, o Conteúdo_C obtido no Ambiente Externo (Conteúdo_{C-E}) pode ser um Objeto de Informação recebido via Correio Eletrônico (mensagens de correio ou notícias dos Canais Filtrados) ou também obtido através de consulta de pesquisa na *Web* (máquinas de busca, bibliotecas digitais, etc). No cenário a seguir, vamos ilustrar um exemplo de como uma Ação de Planejamento ocorre para ajudar o usuário na tarefa de descobrir e sugerir um Conteúdo_{C-E}.

Exemplo de Descoberta e Sugestão de Conteúdo_{C-E}

Considere que o usuário possui o modelo de comportamento ilustrado na Figura 4.20. Num certo instante, o usuário está usando a Ferramenta de Anotação⁶, conforme ilustra a Figura 4.33. Nesta Ferramenta, o usuário primeiro seleciona um trecho de texto e escolhe a categoria da anotação (tela da esquerda), para depois escolher palavras-chave na janela ativada (tela da direita).

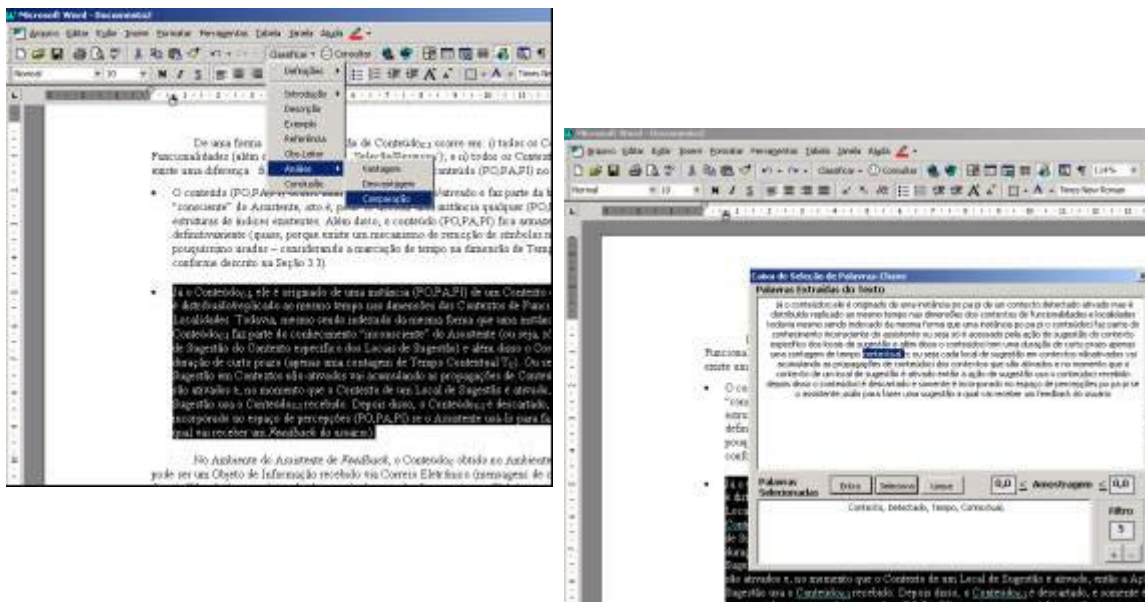


Figura 4.33: Exemplo de Uso da Ferramenta de Anotação

Os planejamentos de ações de sugestão são disparados por várias heurísticas, sendo que uma delas é: ‘a anotação em documentos que estão salvos na mesma pasta do Relatório indica que os Termos/palavras-chaves anotados são fortes indicadores de preferências do usuário’. Visando ajudar o usuário, o Assistente de *Feedback* se baseia no padrão de comportamento do usuário para fazer o plano de ação.

Primeiro simula o comportamento do usuário estando no estado de ação ‘Pesquisa/Seleção’. Neste estado de ação, o Assistente coleta na *Web* novos Conteúdo_{C-E} que contém os Termos recém-anotados. Durante a coleta, o processo de decisão usa a base de conhecimento (PO,PA,PI) para escolher os novos Conteúdo_{C-E} que possuem a maior probabilidade do usuário gostar – dado o contexto de ações prévias do usuário pesquisar na *Web*, ou seja, $P(PI_{=Bom}|PO,PA_{=Pesquisa_na_Web})$. Neste cálculo, os novos Conteúdo_{C-E} escolhidos são aqueles com o maior o número de atributos (das classes CCEL, Figura 4.15) em comum com os PO do contexto de ações prévias de pesquisa na *Web*. Conforme descrito na Definição 2.3.1 (Capítulo 3), nos casos onde não existe a representação de um determinado *Símbolo Composto*, são usados os subconjuntos dos *Símbolos Componentes*, assumindo que eles são condicionalmente independentes. Ou seja, é feita uma aproximação baseando-se nos *Símbolos* que existem, por exemplo, autores, local de origem, área do assunto, etc.

Em seguida, o Assistente de *Feedback* simula o próximo estado de ação no modelo de comportamento do usuário (escolhendo o de maior probabilidade de ocorrer). Digamos que seja o estado de ação ‘Aprendizagem’. Neste estado, o Assistente simula o

⁶ Desenvolvida no âmbito do trabalho [BOR 2000] e disponível para download em www.inf.ufrgs.br/~clermont/assistente_annotacao/

comportamento do usuário fazendo leitura de documentos e anotação de trechos de texto do Conteúdo_{C-E} obtido no estado de simulação anterior. Logo depois, repete o processo de simulação no estado de ação 'Comunicação', que foi o mais provável de ocorrer depois do estado 'Aprendizagem'.

Ao longo desta ação de planejamento/simulação, o Assistente de *Feedback* constrói a indexação do conteúdo Conteúdo_{C-E} rastreado entre estes estados simulados. Desta forma, o Conteúdo Candidato (às Ações de Sugestão) obtido do Ambiente externo vai sendo propagado somente para os Locais de Sugestão mais prováveis de ocorrer, dado o modelo de comportamento do usuário. Da mesma forma que o Conteúdo_{C-I}, o Conteúdo_{C-E} tem a duração de apenas uma contagem de *Tempo Contextual* T_C e somente é incorporado ao espaço de percepções (PO,PA,PI) após o usuário prover *Feedback* (bom ou ruim) para o Local de Sugestão que tiver seu Contexto ativado.

É importante observar que as Ações de Planejamento usam instâncias replicadas da base de conhecimento do SADI que são atualizadas de forma contínua e incremental. Desta forma, se durante o planejamento ocorrem eventos reais (por exemplo, o usuário rejeitou uma sugestão de anotar um conteúdo C) em algum Contexto que participa de uma simulação (neste exemplo, o estado de ação 'Aprendizagem'), então este evento também é enviado para a instância replicada usada na simulação. Uma vez que o conteúdo está rastreado, o *Feedback* do usuário é propagado no modelo simulado, o qual vai se adaptar/corrigir dinamicamente.

Por exemplo, no estado de ação 'Comunicação' da simulação, o Assistente sugere ao usuário enviar um *email* para um colega de equipe, contendo em anexo o Conteúdo_{C-E} (porque na simulação o usuário pesquisou, leu e anotou, e - normalmente quando isto ocorre - o usuário também envia este conteúdo C para o colega). Todavia, na situação real, o usuário rejeitou uma sugestão de anotar C no estado de ação 'Aprendizagem'. Portanto, o Assistente propaga o *Feedback* para o estado de ação 'Comunicação' que é o próximo estado na simulação, de maneira que a sugestão de incluir anexo no *email* considere o evento que ocorreu.

5 CONCLUSÃO

Este trabalho apresenta um modelo eficiente de representação de conhecimento que é construído a partir das relações de causa e efeito entre percepções e ações (onde a eficiência do modelo é baseada na sua escalabilidade, nas operações contínuas e nos custos incrementais). É assumido que é possível perceber o *Ambiente*, é necessário tomar decisões incrementais mediante incerteza, é possível perceber o *feedback* referente ao sucesso ou fracasso das ações escolhidas, e é possível aprender (de forma incremental) com a experiência.

Uma vez que a representação de conhecimento do *Framework SADI* é organizada em vários *Contextos*, quando um padrão é percebido e se ele ocorre em mais de um *Contexto*, então ele é replicado e distribuído usando os serviços SAIC. O *Framework SADI* considera que vai operar em (e inter-operar com) tecnologias de software que também usam o paradigma SAIC, o qual significa basicamente serviços com operações incrementais (independente da quantidade total de dados armazenados) e serviços com crescimento contínuo.

Cada *Contexto*, toda vez que for ativado, realiza sua própria contagem lógica de tempo, definida como *Tempo Contextual* T_C . As percepções entradas num dado *Tempo de Sistema* T_S possuem *Uso Duplo* (*feedforward* e *feedback*). Como o *Tempo Contextual* T_C pode ser diferente do *Tempo de Sistema* T_S , a relação de causa-e-efeito fica relativa a cada contexto ativado.

O ciclo de operação do SADI pode ser implementado como um *pipeline*, e alguns passos podem ser executados em paralelo se eles ocorrem em diferentes *Contextos*. Este processamento paralelo pode ser realizado mesmo se os diferentes *Contextos* possuem *Símbolos* em comum, porque cada *Contexto* possui suas próprias réplicas de *Símbolos*, sua própria semântica, e seu próprio *Tempo Contextual* T_C .

Considerando o *Serviço de Detecção de Contexto* como um todo, ele provê dois serviços básicos: i) O casamento sintático e semântico dos *Símbolos* percebidos; e ii) A organização de *Símbolos* em *Classes que Correspondem a Estruturas de Linguagem (CCEL)* nos *Contextos Estendidos*. O *Framework SADI* preserva as propriedades do *Framework SAIC* mesmo com o uso de *Símbolos Compostos*. O uso do *Modelo de Mundo* (com um conjunto finito de *Estados de Ação* e transições finitas entre estes estados) é potencializado através de *Relacionamentos-Transição* e *Símbolos-Disparadores* – os quais possibilitam um número maior de combinações que são estabelecidas dinamicamente.

Além de reduzir o Problema da Lacuna (Seção 2.8) entre o processo de aprendizagem e o processo de decisão/execução, o *Framework SADI* também minimiza a perda de semântica que cria um *trade-off* (entre a necessidade de coletar dados suficientes; *versus*, rastrear/acompanhar problemas não-estacionários). Estes problemas são reduzidos/minimizados porque as relações de causa-e-efeito estão organizadas em *Contextos* – onde cada um possui:

- Suas próprias réplicas de *Símbolos* e sua própria contagem lógica de *Tempo Contextual* T_C - que é a base para calcular as probabilidades dos *Símbolos*;
 - O T_C é incrementado apenas quando seu *Contexto* é detectado/ativado pelo *Serviço de Detecção de Contextos*. Isto propicia uma técnica de esmaecimento (*fading*) onde: os *Símbolos* antigos e não usados (de *Contextos* mais ativados) tendem a desaparecer mais rápido do que os *Símbolos* antigos e não usados (de *Contextos* menos ativados).
 - O controle semântico dos *Fluxos de Percepção* (PO_{T_C} , PA_{T_C} e PI_{T_C}) provê o significado/importância dos eventos, permitindo que eventos com uma baixa importância estatística (em termos de tamanho da amostra) sejam bastante representativos.

Finalmente, a abordagem do SADI de decisões incrementais e aprendizagem incremental durante a execução, combinada com a sinergia oferecida pelos três relacionamentos simultâneos e úteis (dependência condicional; relacionamentos semânticos multidimensionais; e estrutura de acesso aos dados), tornam o SADI conceitualmente similar a (o que poderia ser referido como) um *Datawarehouse* Orientado a Objetos que opera de forma incremental usando os serviços SAIC. Esta similaridade conceitual se refere a: 1) Dados Orientados a Contexto; 2) Vários Dados sumarizados e sintetizados; 3) Várias estruturas de índices; 4) Representação de Conhecimento Multidimensional; e 5) Armazenamento relativamente grande – proporcional ao escopo do domínio. Nós também acreditamos que esta similaridade conceitual pode ser também aplicada ao cérebro humano – embora este seja muito mais complexo.

5.1 Trabalhos Futuros

A teoria apresentada nesta tese está na fase inicial e ainda necessita ajustes e complementações. A fase de pesquisa envolveu estudos multidisciplinares, e considerando a limitação humana, não foi possível comparar a teoria apresentada com a maioria das principais contribuições já existentes (não citadas devido à impossibilidade de conhecê-las durante o período de estudos).

No âmbito teórico e conceitual, existe uma necessidade de acrescentar novas representações de conhecimento (mais complexas em termos cognitivos) e ainda aprimorar os serviços de inferência e de aprendizagem. O *framework* SADI precisa também de um serviço básico de intercâmbio de conhecimento.

Quanto aos aspectos de implementação, há todo um trabalho de engenharia de software a ser feito - principalmente na modelagem conceitual do *framework*, deixando-o suficientemente flexível para os diversos cenários de tomada de decisão mediante incerteza. Atualmente, o *framework* SADI assume que, durante o serviço de tomada de decisão, as aplicações fazem consultas à base de conhecimento – para escolher a melhor ação a ser aplicada no objeto no contexto em questão. Seria interessante investigar a possibilidade de integrar o mecanismo de tomada de decisão com o ambiente de execução, simplificando a programação das aplicações – ao oferecer o recurso de “sobrecarga probabilística”⁷ de método. Ou seja, dado um mesmo método usado em

⁷ Expressão sugerida pelo Prof. Dr. Marcelo Ladeira (CIC – UnB).

mais de uma classe, a operação de *late bidding* se encarregaria de escolher – além do código possível de ser executado – também qual é o código mais provável de obter êxito no contexto em questão.

Contudo, o *framework* SADI possui um grande potencial de aplicação nos cenários que envolvem coleta de dados massiva e distribuída, e onde existe a necessidade de atualizar de forma contínua e incremental o modelo de comportamento do domínio da aplicação – oferecendo ainda a flexibilidade e o suporte para inferência e aprendizagem em tempo real. Um exemplo de aplicação: serviços de previsão de tempo.

Idealmente, a implementação do *Framework* SADI poderia ser em arquiteturas de computador orientadas a dados (*data-driven*) e orientadas a demanda (*demand-driven*) [TRE 82,GAO 95], com sistemas de *hardware* reconfigurável em tempo de execução (*Reconfigurable Computing* [COM 2002]), e com reconfiguração (em tempo de execução) de sistemas baseados em componentes orientados a objetos (*Component-Oriented Programming* [WCO 2003]), para incorporar (com o recurso da herança múltipla) as novas combinações de *Símbolos* percebidos no *Ambiente*.

REFERÊNCIAS

- [ABO 99] ABOULNAGA, A.; CHAUDHURI, S. Self-Tuning Histograms: Building Histograms Without Looking at Data. **SIGMOD Record**, New York, v.28, n.2, p.181-182, June 1999. Trabalho apresentado na ACM SIGMOD International Conference on Management of Data, 1999, Philadelphia.
- [APP 98] APPARAO, V. et al. **Document Object Model (DOM) Level 1 Specification**. [S.l.]:W3C Recomendaditon, World Wide Web Consortium, 1998. Disponível em: <<http://www.w3.org/TR/REC-DOM-Level-1>>. Acesso em: jun.1999.
- [BAL 2002] BALINSKI, R.; BORGES, C.; GEYER, C.F.R. Canais de Informação com Serviços de Filtragem no Direto. In: FÓRUM INTERNACIONAL DE SOFTWARE LIVRE, 3., 2002, Porto Alegre. **Programação, anais e expositores**. Porto Alegre:SBC,2002. p.38-41.
- [BER 70] BERNE, E. **Sex in Human Loving**. [S.l.]:Pocket Books Publisher, 1970.
- [BOR 97] BORGES, C. **Serviços Básicos para Facilitar o Consumo de Informação num Cenário de Rede**. 1997. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática – UFRGS, Porto Alegre.
- [BOR 98] BORGES, C.; DELIMA, J.V.; COSTA, A.C.R. Services to Aid Information Consumption. In: EUROMEDIA98: MULTIMEDIA, TELEMATICS AND WEB TECHNOLOGY, 1998. **Proceedings...** Leiscester, UK:SCS, 1998. p.50-54.
- [BOR 99] BORGES, C. **Um Estudo Sobre Metadados na Web, Representação e Intercâmbio de Conhecimento em Agentes**. 1999. Trabalho Individual (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [BOR 2000] BORGES, C. *et al.* Assistant Agent for Reading and Annotating. In: ICSC SYPOSIA ON INTELLIGENT SYSTEMS & APPLICATIONS, 2000, Wollongong. **Proceedings...** Canada: ICSC Academic, 2000.
- [BOR 2003a] BORGES, C. ; GEYER, C.F.R. Estratégias de Governo para Promover o Desenvolvimento de Software Livre. **Revista RESI**, Campo Largo, v.2, n.1, jul. 2003.
- [BOR 2003b] BORGES, C. ; GEYER, C.F.R. **The Organization of the Virtual Space with Privacy Protection and Security Control**. Workshop – Fórum Social Mundial, Porto Alegre, jan. 2003. Disponível em: <<http://www.inf.ufrgs.br/~clermont/OrgVirtSpace.html>>. Acesso em: nov. 2004.
- [BRA 98] BRAY,T.; PAOLI, J. ; SPERBERG-MCQUEEN C.M. Extensible Markup Language (XML) 1.0. [S.l.]:W3C Recomendaditon, World Wide Web Consortium, 1998. Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: jun.1999.
- [CAR 99] CARNEIRO, A.L. ; DA SILVA, W.T. **Introdução às Redes Bayesianas**.

- Relatório de Pesquisa CIC/UnB-09/99*, 1999. Disponível em: <ftp://ftp.cic.unb.br/pub/publications/report/rr.99-09.ps.Z>. Acesso em: maio 2000.
- [CAS 97] CASTILLO, E.; GUTIERREZ, J.M.; HADI, A.S. **Expert Systems and Probabilistic Network Models**. New York: Springer-Verlag, 1997.
- [CDF 97] CHANNEL DEFINITION FORMAT (CDF). Microsoft Corporation. 1997. Submission Request to W3C. Disponível em: <http://www.w3.org/TR/NOTE-CDFsubmit.html>. Acesso em: jun.1999.
- [CHA 99] CHANDRASEKARAN, B.; JOSEPHSON, J.R. ; BENJAMINS, V.R. What are Ontologies, and Why Do We Need Them?. **IEEE Intelligent Systems**, [S.l.], p.20-26, Jan./Feb. 1999.
- [CHI 94] CHICKERING, M.D.; GEIGER, D. ; HECKERMAN, D. **Learning Bayesian Networks is NP-Hard**. Technical Report. 1994. Microsoft Research. Disponível em: <ftp://ftp.research.microsoft.com/pub/tr-94-17.ps>. Acesso em: jan. 2000.
- [COM 2002] COMPTON, K. ; HAUCK, S. Reconfigurable Computing: A Survey of Systems and Software. **ACM Computing Surveys**, New York, v.34, n.2, p. 171–210, June 2002.
- [CRU 2003] CRUSE, H. The Evolution of Cognition – a Hypothesis. **Elsevier Cognitive Science**, [S.l.], v.27, p.135-155, 2003.
- [DAV 77] DAVIS, R.; BUCHANAN, B. ; SHORTLIFFE, E. Production Rules as a Representation for a Knowledge-Based Consultation Program. **Artificial Intelligence**, Amsterdam, v.8, p.15-45, 1977.
- [DEL 74] PROBABILIDADE. Grande Enciclopédia Delta Larousse. Rio de Janeiro: Ed. Delta, 1974. p. 5546.
- [DOY 79] DOYLE, J. A Truth Maintenance System. **Artificial Intelligence**, Amsterdam, v.12, p. 231-272, 1979.
- [DUV 99] DUVAL, E.; HODGINS, W. ; WASON, T. **Instructional Management Systems Project**. 1999. Disponível em: <http://www.imsproject.org/metadata>. Acesso em: jun.1999.
- [ELD 96] ELDER, J. F. IV ; PREGIBON, D A Statistical Perspective on Knowledge Discovery in Databases. In: FAYYAD, U.M. *et al.* (Ed.). **Advances in Knowledge Discovery and Data Mining**. [S.l.]: AAAI Press, 1996.
- [FIN 92] FINNIN, T.; MCKAY, D. ; FRITZSON, R. **An Overview of KQML: A Knowledge Query and Manipulation Language**. 1992. Disponível em: <http://www.cs.umbc.edu/kqml/>. Acesso em: jun.1999.
- [FIP 2004] FIPA - FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. Disponível em: <http://www.fipa.org>. Acesso em: 2004.
- [FRA 99] FRANK, G. ; FARQHAR, A. ; FIKES, R. Building a Large Knowledge Base from a Structured Source. **IEEE Intelligent Systems**, [S.l.], p. 47-54, Jan./Feb. 1999.
- [FRI 2002] FRIGHETTO, M. **Regras de Associação Aplicadas aos Filtros de Mensagens e Canais de Informação do Projeto Direto**. 2002. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [GHA 98] GHAHRAMANI, Z. Learning Dynamic Bayesian Networks. In: INTERNATIONAL SUMMER SCHOOL ON NEURAL NETWORKS, 1997, Vietri Sul Mare, It. **Adaptive Processing of Sequences and Data Structures: tutorial lectures**. Berlin: Springer-Verlag, 1998. p.168-197. (Lecture Notes in Computer Science, v.1387). Disponível em: <http://citeseer.ist.psu.edu/ghahramani97learning.html>. Acesso em: mar. 2000.

- [HEC 86] HECKERMAN, D. Probability Interpretation for MYCIN's Certainty Factors. In: KANAL, L.N.; LEMMER, J.F. (Ed.). **Uncertainty in Artificial Intelligence**. Amsterdam: Elsevier, 1986. p. 167-196.
- [HEC 98] HECKERMAN, D. ; HORVITZ, E. Inferring Informational Goals from Free-Text Queries – A Bayesian Approach. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 14., 1998. **Proceedings...** Madison, WI:[s.n.], 1998. p. 230-237.
- [HOR 97] HORVITZ, E. Agents with Beliefs: Reflections on Bayesian Methods for User Modeling. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 6., 1997, Sardinia, It. **User Modeling: proceedings**. Sardinia: Springer-Verlag, 1997.
- [HUN 98] HUNTER, A.; PARSONS, S. (Ed.). **Applications of Uncertainty Formalisms**. Berlin:Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v.1455).
- [JEN 2001] JENSEN, F.V. **Bayesian Networks and Decision Graphs**. [S.l.]: Springer-Verlag, 2001.
- [KLE 86] KLEER, J. An Assumption-Based TMS. **Artificial Intelligence**, Amsterdam, v.28, n.127-162, 1986.
- [KOL 97] KOLLER, D. ; PFEFFER, A. Object-oriented Bayesian networks. In: INTERNATIONAL CONFERENCE OF UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, UAI, 1997. Disponível em: <<http://citeseer.ist.psu.edu/koller97objectoriented.html>>. Acesso em: out. 2003.
- [KOR 89] KORTH, H.F. ; SILBERSCHATZ. **Sistema de Banco de Dados**. Tradução de Database System Concepts, revisão técnica Routh Terada. São Paulo: MacGraw-Hill, 1989.
- [LAB 99] LABROU, Y.; FININ, T. ; PENG, Y. Agent Communication Languages: The Current Landscape. **IEEE Intelligent Systems**, [S.l.], p. 45-52, Mar./Apr. 1999.
- [LAD 2000] LADEIRA, M. **Diagrama de Influências Múltiplo Secionado**. 2000. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- [LAN 2001] LANGSETH, H. ; BANGS, O. Parameter Learning in Object Oriented Bayesian Networks. **Annals of Mathematics and Artificial Intelligence**, Dordrecht, n.31, v.1/4, p. 221-243, 2001. Disponível em: <<http://citeseer.ist.psu.edu/langseth01parameter.html>>. Acesso em out.2003.
- [LAN 2003] LANGSETH, H. ; NIELSEN, T.D. **Fusion of Domain Knowledge with Data for Structural Learning in OO Domains**. Disponível em: <<http://www.cs.auc.dk/research/DSS/papers/technial.ps>>. Acesso em: out. 2003.
- [LAS 99] LASSILA, O ; SWICK, R. **Resource Description Framework (RDF) Model and Syntax Specification**. [S.l.]: W3C Recommendation. World Wide Web Consortium, 1999. Disponível em: <<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>>. Acesso em: jun.1999.
- [LIP 72] LIPSCHULTZ, S. **Teoria e Problemas de Probabilidade**. São Paulo: McGraw-Hill, 1972.
- [LOP 99] LOPES, P.A. **Probabilidades e Estatística**. [S.l.]:Reichmann &Affonso Ed., 1999.
- [MAC 2003] MACHADO, E.F.N. **Metapsicologia: Princípios para um Conceito Alternativo de Cognição em Inteligência Artificial**. 2003. Dissertação

- (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [MEL 2002] MELLO, L. C. **Um Assistente de Feedback para o Serviço de Filtragem do Software Direto**. 2002. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [NIL 98] NILSSON, N. **Artificial Intelligence – A New Synthesis**. San Francisco, CA: Morgan Kaufmann, 1998.
- [PAL 2000] PALAZZO, L.A.M. **Modelos Proativos para Hipermídia Adaptativa**. 2000. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- [PEA 88] PEARL, J. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**. [S.l.]:Morgan Kaufmann Publishers, 1988.
- [PEA 2000] PEARL, J. **Causality – Models, Reasoning and Inference**. Cambridge: Cambridge University Press, 2000.
- [PHI 91] PHIPPS, G., DERR, M.A. ; ROSS, K.A. Glue-Nail: a Deductive Database System. In: ACM-SIGMOD CONFERENCE ON MANAGEMENT OF DATA, 1991. Disponível em: <<http://citeseer.nj.nec.com/phipps91gluenail.html>>. Acesso em: out. 2003.
- [REI 80] REITER, R. A Logic for Default Reasoning. **Artificial Intelligence**, Amsterdam, v.13, n.12, p.81-132, Apr. 1980.
- [RIC 93] RICH, E. ; KNIGHT, K. **Inteligência Artificial**. 2.ed. São Paulo: Makron Books, 1993.
- [RUS 95] RUSSELL, S. ; NORVIG, P. **Artificial Intelligence – A Modern Approach**. New Jersey: Prentice Hall, 1995.
- [SEA 95] SEARLE, J.R. **Intencionalidade**. São Paulo: Martins Fontes, 1995.
- [SUT 99] SUTTON, R.S ; BARTO, A.G. **Reinforcement Learning – An Introduction**. London: MIT Press, 1999.
- [SUT 2003] SUTTON, C. *et al.* Guided Incremental Construction of Belief Networks. In: INTERNATIONAL SYMPOSIUM ON INTELLIGENT DATA ANALYSIS, 5., 2003, Berlin. Disponível em: <<http://citeseer.nj.nec.com/sutton03guided.html>>. Acesso em: out. 2003.
- [SWA 99] SWARTOUT, W. ; TATE, A. Ontologies. **IEEE Intelligent Systems**, Los Alamitos, p.18-19, Jan./Feb. 1999.
- [TRE 82] TRELEAVEN, P.C.; BROWNBRIDGE, D.R.; HOPKINS R.P. Data-Driven and Demand-Driven Computer Architecture. **ACM Computing Surveys**, New York, v.14, n.1, p. 93-143, March 1982.
- [WCO 2003] INTERNATIONAL WORKSHOP ON COMPONENT-ORIENTED PROGRAMMING, WCOP, ECOOP, 2003, Darmstadt, Germany. Disponível em: <<http://research.microsoft.com/~cszypers/events/WCOP2003/>>. Acesso em: out. 2003.
- [WON 2001] WONG, S.K.M. An extended relational data model for probabilistic reasoning. **Journal of Intelligent Information Systems**, Dordrecht, n.9, p.181-202, 1997.
- [WON 2002] S.K.M. WONG, D WU ; C.J. BUTZ. Triangulation of Bayesian networks: a Relational Database Perspective. In: INTERNATIONAL CONFERENCE ON ROUGH SETS AND CURRENT TRENDS IN COMPUTING, RSCTC, 2002. Disponível em: <<http://citeseer.ist.psu.edu/567539.html>>. Acesso em: out. 2003.
- [ZAD 78] ZADEH, L.A. Fuzzy Sets as a Basis for a Theory of Possibility. **Fuzzy Sets and Systems**, Amsterdam, v.1, p.2-28, 1978.

