

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

MÁRCIA CRISTINA MORAES

**Agentes Improvisacionais
como Agentes Deliberativos**

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

Prof. Dr. Antônio Carlos da Rocha Costa
Orientador

Porto Alegre, janeiro de 2004.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Moraes, Márcia Cristina.

Agentes Improvisacionais como Agentes Deliberativos/
Márcia Cristina Moraes. – Porto Alegre:Programa de Pós-
Graduação, 2004.

214f.:il.

Tese (doutorado) – Universidade Federal do Rio Grande
do Sul. Programa de Pós-Graduação, Porto Alegre, BR-RS, 2004.
Orientador: Antônio Carlos da Rocha Costa.

1. Agentes improvisacionais. 2. Agentes deliberativos. 3.
Direção improvisacional. 4. Inteligência artificial distribuída. 5.
Sistemas multiagentes e coordenação. I. Costa, Antônio Carlos da
Rocha. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitora Adjunta de Pós-Graduação: Prof^a. Jocélia Grazia

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

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

À minha família

AGRADECIMENTOS

Gostaria de iniciar este documento agradecendo ao meu orientador, Prof. Dr. Antônio Carlos da Rocha Costa pelos vários anos, mais especificamente sete contando mestrado e doutorado, de inquietações, questionamentos e aconselhamentos que contribuíram para a construção desta tese. Fico muito feliz em poder ter trabalhado com uma pessoa com tamanho conhecimento e sempre disposta a me incentivar durante esta longa caminhada. Acredito que esta caminhada, apesar de longa, é o início de vários caminhos que ainda podemos trilhar em conjunto.

Agradeço a minha família pelo apoio e incentivo para realização deste projeto e pelo entendimento das minhas faltas em alguns almoços, jantas e outras atividades familiares. Agradeço em particular aos meus pais, Roque Moraes e Ilza Maria Beiler Moraes, pelo exemplo e educação que me tornaram a pessoa que sou hoje. Agradeço aos meus irmãos Ana Cláudia Moraes e Rodrigo Moraes por terem me “aturado” nos dias de mau humor e ao meu esposo Paulo Ricardo Tatsch Dutra que esteve sempre ao meu lado, abrindo mão de alguns finais de semana na praia, idas ao cinema, etc, para ficar comigo em casa, me alegrando e me dando esperanças nas horas difíceis.

Agradeço aos meus amigos em especial a Patrícia Augutin Jaques, Ana Carolina Bertoletti e Rejane Frozza pelas conversas e trocas de idéias.

Agradeço ao CNPq pelo incentivo a esta tese através de uma bolsa de doutorado de quatro anos.

Agradeço a Capes, programa PDEE/UFRGS, pela bolsa de doutorado sanduíche que possibilitou o meu intercâmbio com a Dra. Barbara Hayes-Roth, criadora do conceito de direção improvisada e agentes improvisacionais. Este intercâmbio foi bastante importante para o desenvolvimento de algumas das idéias apresentadas nesta tese.

Por fim, agradeço a todos que contribuíram de alguma maneira para a realização desta tese.

SUMÁRIO

LISTA DE ABREVIATURAS.....	10
LISTA DE FIGURAS.....	11
LISTA DE TABELAS.....	13
RESUMO.....	15
ABSTRACT.....	16
1 INTRODUÇÃO	17
2 MOTIVAÇÕES E CONTRIBUIÇÕES ESPERADAS	19
3 TRABALHOS RELACIONADOS	22
3.1 Projeto Teatro Virtual	22
3.2 Empresa Extempo Systems Inc.	25
3.3 Projeto Oz.....	28
3.4 Projeto Improv NYU.....	33
3.5 Considerações sobre o Capítulo e Trabalhos Anteriores em Direção Improvisada 36	
4 TEATRO IMPROVISACIONAL COMO FUNDAMENTO PARA AGENTES IMPROVISACIONAIS	38

4.1	Elementos do Teatro Improvisacional.....	39
4.1.1	O Texto	40
4.1.2	O Ator	40
4.1.3	O Público	41
4.2	Agentes Improvisacionais	41
4.2.1	Improvisação Dirigida	41
4.2.2	Características dos Agentes Improvisacionais	43
4.3	Considerações sobre o Capítulo	44
5	PAPEL DO DIRETOR NO TEATRO IMPROVISACIONAL.....	46
5.1	Características de um Diretor Criativo ou Improvisacional.....	46
5.2	Construção do Script.....	47
5.3	Avaliação	48
5.4	Resolução de Problemas.....	48
5.5	Considerações sobre o Capítulo	48
6	TIPOS DE IMPROVISACÃO	50
6.1	Categorização da Improvisação em Cada um dos Tipos de Improvisação.....	51
6.1.1	Improvisação de Texto	52
6.1.2	Improvisação da Personalidade	52
6.1.3	Improvisação do Público	53
6.1.4	Improvisação de Solução de Problemas	53
7	DIREÇÃO COMO COORDENAÇÃO EM SISTEMAS IMPROVISACIONAIS	55
7.1	Coordenação em Sistemas Multiagentes	55
7.2	Cooperação.....	57
7.3	Competição.....	62
7.3.1	Negociação	62
7.4	Espaço de Formas de Coordenação	65
7.4.1	Eixo dos Estilos	65
7.4.2	Eixo das Técnicas	67
7.5	Visão Geral das Formas de Coordenação e sua Classificação no Espaço de Coordenação	68

7.5.1 Huang e Sycara [HUA 2002]	68
7.5.2 Kraus [KRA 2001]	68
7.5.3 Ortiz et al [ORT 2000]	69
7.5.4 Guttman e Maes [GUT 98a] [GUT 98b]	70
7.5.5 Wurman, Wellman e Walsh [WUR 98]	71
7.5.6 Grupo de Lesser [SAN 95] [JEN 99] [ZHA 2000] [ZHA 2001].....	71
7.5.7 Zlotkin e Rosenchein [ZLO 89] [ZLO 90] [ZLO 96a] [ZLO 96b] [ROS 97]	72
7.5.8 Jennings et al [JEN 2000] [JEN 2001]	73
7.5.9 Bonarini e Restelli [BON 2002]	75
7.5.10 Rede de Contrato [HUH 99].....	76
7.5.11 Sistemas de Quadro Negro [HUH 99].....	77
7.5.12 Planejamento Distribuído [DUR 99].....	77
7.5.13 Análise do Espaço de Formas de Coordenação.....	79
7.6 Classificação dos Trabalhos relacionados no Espaço de Formas de Coordenação	81
7.6.1 Teatro virtual	81
7.6.2 Empresa Extempo Systems Inc.	82
7.6.3 Projeto Oz.....	82
7.6.4 Projeto Improv.....	82
7.7 Espaço de Formas de Coordenação Estendido	82
7.7.1 Eixo da Distribuição da Coordenação	83
7.8 Considerações sobre o Capítulo	84
8 AGENTES DELIBERATIVOS BASEADOS EM ARQUITETURA	85
BDI.....	85
8.1 Aspectos Relacionados à Teoria de Raciocínio Prático.....	86
8.2 Aplicações da Teoria de Raciocínio Prático: Arquiteturas BDI	88
8.2.1 Arquitetura IRMA	88
8.2.2 Arquitetura dMARS	92
8.2.3 Esquema de Wooldridge para uma Arquitetura BDI Genérica	95
8.2.4 Arquitetura BDI para Intenções Conjuntas	99
8.2.5 Arquitetura BDI com Restrições	104
8.3 Lógicas BDI.....	107
8.4 Considerações sobre o Capítulo	109
9 IDENTIFICAÇÃO DE TIPOS DE IMPROVISACÃO IMPLÍCITA	110
NAS ARQUITETURAS BDI	110
9.1 Arquitetura IRMA	110

9.1.1	Improvisação de Texto	110
9.1.2	Improvisação de Público	111
9.1.3	Improvisação de Resolução de Problemas	111
9.2	Arquitetura dMARS	111
9.2.1	Improvisação do Texto	111
9.2.2	Improvisação de Resolução de Problemas	112
9.3	Diagrama Esquemático de Wooldridge para Arquitetura BDI Genérica.....	112
9.3.1	Improvisação de Texto	112
9.3.2	Improvisação de público.....	113
9.3.3	Improvisação de Resolução de Problemas	113
9.4	Arquitetura BDI para Intenções Conjuntas	113
9.4.1	Improvisação do Texto	113
9.4.2	Improvisação da Solução de Problemas	114
9.5	Arquitetura BDI com Restrições.....	114
9.5.1	Improvisação do Texto	114
9.5.2	Improvisação de Solução de Problemas	115
9.6	Considerações sobre o Capítulo	115
10	ARQUITETURA BDI IMPROVISACIONAL	116
10.1	Diagrama Esquemático da Arquitetura BDI Improvisacional	116
10.2	Mapeamento da Terminologia de Crenças, Desejos e Intenções da Arquitetura BDI Tradicional para Arquitetura BDI Improvisacional.....	118
10.3	Improvisação de Solução de Problemas por Analogia.....	119
10.3.1	Analogia	119
10.3.2	Arquitetura e Funcionamento do Módulo de Improvisação de Solução de Problemas	
	127	
10.4	Definição de Agentes Improvisacionais como Agentes Deliberativos.....	134
10.5	Considerações sobre o Capítulo	135
11	APLICAÇÃO DA ARQUITETURA BDI IMPROVISACIONAL PARA UMA ARQUITETURA DE GUIAS VIRTUAIS	136
11.1	Tipos de Coordenação Improvisacional Presentes na Arquitetura	138
11.2	Interação entre os Componentes da Arquitetura	138
11.2.1	Interação entre os Componentes da Arquitetura para o Papel de Diretor	138
11.2.2	Interação entre os Componentes da Arquitetura para o Papel de Ator.....	139

11.3 Módulo de Aquisição de Conhecimento	140
11.4 Improvisação de Texto: Construção da Intenção como Script Abstrato de Comportamento (SAC)	144
11.4.1 Visão de Planos como Programas versus Planos como Intenções	145
11.4.2 Intenção, Improvisação e Satisfação de Restrições	146
11.4.3 Papéis dos Agentes e Tipos de Improvisação do Texto	148
11.4.4 Classes e Tipos de Restrições	149
11.4.5 Funcionamento do Módulo para o Papel de Diretor	149
11.4.6 Funcionamento do Módulo para o Papel de Ator	156
11.4.7 Funcionamento do Módulo para o Papel de Diretor-Ator	161
11.5 Improvisação para Solução de Problemas	162
11.5.1 Funcionamento do Módulo para o Papel de Diretor	162
11.5.2 Funcionamento do Módulo para o Papel de Ator	168
11.5.3 Funcionamento do Módulo para o Papel de Diretor-Ator	169
11.6 Módulo de Percepção	169
11.7 Módulo de Ação	170
12 ESTUDO DE CASO: FUNCIONAMENTO DA ARQUITETURA DE AGENTES IMPROVISACIONAIS PARA O MUSEU VIRTUAL SAGRES	171
12.1 Módulo de Aquisição de Conhecimento	171
12.1.1 Considerações quanto aos Tipos de Improvisação que Podem Ocorrer Durante esta Apresentação	176
12.2 Módulo de Construção da Intenção	176
12.2.1 Execução para o Papel de Diretor	177
12.2.2 Execução para o Papel de Ator	182
12.3 Módulo de Improvisação de Solução de Problemas	184
12.3.1 Funcionamento para o Papel de Ator	184
12.3.2 Funcionamento para o Papel de Diretor	187
12.4 Considerações sobre o Capítulo	194
13 CONCLUSÕES	196
13.1 Trabalhos Futuros	197
REFERÊNCIAS	199

LISTA DE ABREVIATURAS

ACL	<i>Agent Communication Language</i>
BDI	<i>Belief-Desire-Intention</i>
CBR	<i>Case-Based Reasoning</i>
CLP	<i>Constraint Logic Problem</i>
CSDL	<i>Cooperative System Design Language</i>
DOC	Domínios Orientados a Custo
DOE	Domínios Orientados a Estados
DOT	Domínios Orientados a Tarefas
dMARS	Sistema de Raciocínio Multiagente
FIPA	<i>Foundation for Intelligent Physical Agents</i>
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
ICDK	<i>Imp Character Development Kit</i>
IHC	Interação Homem-Computador
IRMA	<i>Intelligent Resource-Bounded Machine Architecture</i>
KQML	<i>Knowledge Query and Manipulation Language</i>
PCL	<i>Proteus Configuration Language</i>
PRC	Protocolo de Rede de Contrato
PRS	<i>Practical Reasoning System</i>
PSRD	Problemas de Satisfação de Restrições Distribuídas
SMA	Sistemas Multiagentes
SAC	<i>Script Abstrato de Comportamento</i>
TUMA	Teoria de Utilidade Multiatributo

LISTA DE FIGURAS

Figura 3.1: Arquitetura dos Agentes	24
Figura 3.2: Interface do Modo de Bonecos Animados disponível em www-ksl.stanford.edu/cait/demos/woggles.html	25
Figura 3.3: Arquitetura baseada em componentes para personagens interativos	26
Figura 3.4: Interface da ferramenta ICDK para criação de tópicos de interação com o usuário	27
Figura 3.5: Arquitetura Geral do Projeto OZ	28
Figura 3.6: Arquitetura detalhada do projeto Oz [KEL 91]	29
Figura 3.7: Arquitetura do Tok [BAT 92a]	30
Figura 3.8: Arquitetura básica para um ator no projeto Improv	33
Figura 3.9: Exemplo de script associado a regras de decisão	34
Figura 3.10: Exemplo da definição da personalidade do ator no momento de sua criação ..	35
Figura 3.11: Comunicação entre agentes no Improv	35
Figura 7.1: Taxonomia de como os agentes podem coordenar suas ações e comportamentos	56
Figura 7.2: Tipologia de cooperação e seu enquadramento em sistemas multiagente	58
Figura 7.3: Categorias do Eixo dos Estilos	66
Figura 7.4: Categorias das Técnicas	67
Figura 7.5: Espaço de Formas de Coordenação	78
Figura 7.6: Espaço de Coordenação com Domínios Predominantes	79
Figura 7.7: Eixos do Espaço de Formas de Coordenação Estendido	83
Figura 7.8: Visão inicial do eixo da distribuição	84
Figura 8.1: Uma arquitetura para agentes limitados a recursos [BRA 88]	89
Figura 8.2: Arquitetura IRMA [POL 92]	90
Figura 8.3: Visão Geral da Arquitetura dMARS [D'IN 97]	92
Figura 8.4: Esquema de uma Arquitetura BDI Genérica	95
Figura 8.5: Laço de Controle de um Agente BDI	98
Figura 8.6: Detalhamento do Algoritmo do Laço de Controle do Agente BDI	98
Figura 8.7: Arquitetura Funcional do Agente – Intenções Dirigidas ao Futuro	101
Figura 8.8: Arquitetura Funcional do Agente – Intenções dirigidas ao Presente	103
Figura 8.9: Arquitetura do Agente	104
Figura 8.10: Laço Principal BDI [CHA 2001]	105
Figura 10.1: Arquitetura BDI Improvisacional como uma extensão do diagrama esquemático de Wooldridge	117
Figura 10.2: Laço de Controle de um Agente BDI Improvisacional	118
Figura 10.3: Analogia Transformacional (retirada de [RIC 93])	122
Figura 10.4: Analogia Derivacional (retirada de [RIC 93])	123
Figura 10.5: Modelo genérico de objetos	128
Figura 10.6: Arquitetura do Módulo de Improvisação para Solução de Problemas	129
Figura 10.7: Algoritmo de alto nível para submódulo identificação do tipo de improvisação	130

Figura 10.8: Algoritmo de alto nível do submódulo de construção da analogia	133
Figura 10.9: Algoritmo de alto nível para submódulo de transformação da analogia em intenção.....	134
Figura 11.1: Arquitetura Genérica de Guias Virtuais usando Arquitetura BDI Improvisacional	137
Figura 11.2: Interação entre os componentes da arquitetura para o papel de diretor	139
Figura 11.3: Interação entre os componentes da arquitetura para o papel de ator	140
Figura 11.4: Interação entre diretor e escritor na aquisição de conhecimento	141
Figura 11.5: Algoritmo de alto nível do módulo de aquisição de conhecimento	141
Figura 11.6: Protocolo para aquisição de conhecimento	142
Figura 11.7: Exemplos de tipos, áreas e subáreas de conhecimento	144
Figura 11.8: Arquitetura genérica do módulo de construção de intenções	145
Figura 11.9: Intenção do agente no seu papel de diretor	150
Figura 11.10: Exemplos de ações e restrições relacionadas no módulo de construção de <i>scripts</i> para o papel de diretor	151
Figura 11.11: Arquitetura de módulo da construção do script para o papel de diretor	152
Figura 11.12: Transformação de uma regra se..então em uma estrutura de intenção como <i>script</i> abstrato de comportamento	153
Figura 11.13: Algoritmo de alto nível do escalonador de atividades para o papel do diretor	155
Figura 11.14: Algoritmo de alto nível do escalonador de comportamento para o papel de diretor	156
Figura 11.15: Exemplo genérico da intenção de um ator	157
Figura 11.16: Exemplos de ações e restrições de ordem relacionadas no módulo de construção de <i>scripts</i> para o papel de ator.....	158
Figura 11.17: Exemplos de ações e restrições de comportamento relacionadas no módulo de construção de <i>scripts</i> para o papel de ator.....	158
Figura 11.18: Arquitetura do módulo de construção da intenção para o papel de ator	159
Figura 11.19: Algoritmo de alto nível do escalonador de atividades para o papel de ator.	160
Figura 11.20: Algoritmo de alto nível do escalonador de comportamento para o papel de ator	161
Figura 11.21: Arquitetura do módulo de improvisação para solução de problemas para o papel de diretor	163
Figura 11.22: Algoritmo de alto nível da identificação do tipo de improvisação para a arquitetura de guias virtuais.....	164
Figura 11.23: Algoritmo de alto nível da construção da analogia para a arquitetura de guias virtuais	167
Figura 11.24: Algoritmo de alto nível da transformação da analogia em intenção para a arquitetura de guias virtuais.....	168
Figura 12.1: Inclusão da atividade inicial na intenção	177
Figura 12.2: Inclusão da atividade antes dos dinossauros na intenção.....	178
Figura 12.3: Inclusão das atividades vestígios de dinossauros, um novo dinossauro em Santa Maria e o passado na intenção	180
Figura 12.4: Estrutura Final da intenção para objetivo dinossauros.....	181
Figura 12.5: Atualização da intenção através da improvisação de meio.....	189
Figura 12.6: Atualização da intenção através da improvisação de objetivo sem efeito conhecido.....	191

LISTA DE TABELAS

Tabela 6.1: Tipos de Improvisação.....	51
Tabela 6.2: Tipos de Eventos e Mecanismos para Solucioná-los.....	53
Tabela 7.1: Número de trabalhos em cada domínio	79
Tabela 10.1: Comparação entre as Etapas da Analogia e do Raciocínio Baseado em Casos	127
Tabela 11.1: Estrutura da base de informação do curso de ação	142
Tabela 11.2: Estrutura da base de informação da atividade	143
Tabela 11.3: Estrutura da base de informação do conteúdo	143
Tabela 11.4: Estrutura de uma percepção do agente diretor.....	162
Tabela 12.1: Estrutura da base de informação do conteúdo para o SAGRES.....	172
Tabela 12.2: Estrutura de armazenamento do curso de ação para uma apresentação sobre dinossauros	172
Tabela 12.3: Estrutura de armazenamento da atividade apresentação do guia	172
Tabela 12.4: Estrutura de armazenamento do conteúdo apresentação do guia	173
Tabela 12.5: Estrutura de armazenamento da atividade antes dos dinossauros	173
Tabela 12.6: Estrutura de armazenamento do conteúdo antes dos dinossauros	173
Tabela 12.7: Estrutura de armazenamento da atividade vestígios de dinossauros	173
Tabela 12.8: Estrutura de armazenamento do conteúdo vestígios de dinossauros.....	174
Tabela 12.9: Estrutura de armazenamento da atividade um novo dinossauro em Santa Maria	174
Tabela 12.10: Estrutura de armazenamento do conteúdo um novo dinossauro em Santa Maria.....	174
Tabela 12.11: Estrutura de armazenamento da atividade o passado	174
Tabela 12.12: Estrutura de armazenamento do conteúdo o passado	175
Tabela 12.13: Estrutura de armazenamento da atividade o presente.....	175
Tabela 12.14: Estrutura de armazenamento do conteúdo o presente	175
Tabela 12.15: Estrutura de armazenamento da atividade finalizar apresentação	175
Tabela 12.16: Estrutura de armazenamento do conteúdo finalizar apresentação.....	176
Tabela 12.17: Atualização da estrutura de armazenamento do curso de ação para uma apresentação sobre dinossauros	176
Tabela 12.18: Informações adicionais sobre conteúdo sítio arqueológico	185
Tabela 12.19: Formato da percepção de falha para o conteúdo sítio arqueológico.....	185
Tabela 12.20: Informações adicionais sobre conteúdo evolução humana e dos primatas .	186

Tabela 12.21: Formato da percepção de falha para o conteúdo evolução humana e dos primatas	186
Tabela 12.22: Informações adicionais sobre conteúdo evolução humana e dos primatas .	186
Tabela 12.23: Formato da percepção de falha para o conteúdo meteoritos	187
Tabela 12.24: Descrição do objeto problema para sítios arqueológicos	187
Tabela 12.25: Estrutura de armazenamento da atividade sítio arqueológico	188
Tabela 12.26: Estrutura de armazenamento do conteúdo sítio arqueológico	188
Tabela 12.27: Descrição do objeto problema para meteoritos	189
Tabela 12.28: Estrutura de armazenamento da atividade minerais	190
Tabela 12.29: Estrutura de armazenamento do conteúdo minerais	190
Tabela 12.30: Descrição do objeto problema para evolução humana e dos primatas	192
Tabela 12.31: Estrutura de armazenamento do curso de ação para uma apresentação sobre estágios da terra	192
Tabela 12.32: Estrutura de armazenamento da atividade evolução humana e dos primatas	193
Tabela 12.33: Estrutura de armazenamento do conteúdo evolução humana e dos primatas	193
Tabela 12.34: Estrutura de armazenamento da atividade finalizar apresentação para curso de ação estágios da terra	193
Tabela 12.35: Estrutura de armazenamento do conteúdo finalizar apresentação para curso de ação estágios da terra	193

RESUMO

Improvisação tem sido considerada uma característica importante para agentes que pretendem operar de maneira consistente com a situação do momento, exibindo um comportamento credível e interessante. A improvisação deve estar presente tanto nos agentes individuais quanto nas sociedades de agentes. Desta maneira, esta tese irá abordar estes dois aspectos da improvisação.

Propomos a visão de que, agentes capazes de realizar improvisação, os agentes improvisacionais, são um tipo de agente deliberativo capaz de solucionar problemas por improvisação. Neste sentido, buscamos identificar dentro de uma arquitetura clássica de agentes deliberativos, a arquitetura BDI (*belief-desire-intention*), a existência e/ou a possibilidade da inclusão de componentes de improvisação nesta arquitetura.

Para resolver problemas complexos, estes agentes precisam estar agrupados em sociedades e estas sociedades, por sua vez, precisam produzir comportamentos coerentes. A coordenação é a área da Inteligência Artificial responsável por este objetivo. Propomos que a coordenação de agentes que improvisam pode ser realizada por meio de um processo de direção improvisacional, no sentido usado no contexto do teatro improvisacional.

Ao longo deste documento, iremos mostrar nosso entendimento sobre agentes improvisacionais como agentes deliberativos e coordenação como direção improvisacional. Com isto, defende-se nesta tese que o uso da improvisação em agentes improvisacionais possibilita que os agentes improvisem comportamentos interativos, de maneira coerente, melhorando seu desempenho como solucionadores de problemas, criando e mantendo uma ilusão de vida para os agentes interativos e contribuindo para o aperfeiçoamento dos sistemas multiagentes.

Palavras-chave: Agentes Improvisacionais, Agentes Deliberativos, Direção Improvisacional, Inteligência Artificial Distribuída, Sistemas Multiagentes e Coordenação.

Improvisational Agents as Deliberative Agents

ABSTRACT

Improvisation has been considered an important characteristic for agents that intend to act in accordance to a current situation, showing believable and interesting behavior. Improvisation should be present in two cases: individual agents and society of agents. In this way, this thesis is going to approach these two aspects of improvisation.

We propose that, agents capable of improvisation, improvisational agents, are one kind of deliberative agents able to solve problems of improvisation. In our study, we try to identify the existence and/or the possibility to include improvisation components in a BDI (belief-desire-intention) architecture.

To solve complex problems these agents have to be grouped in societies and these societies have to express coherent behaviors. Coordination is the area of artificial intelligence responsible for this subject. As a result of our study we argue that the coordination of agents that improvise can be done through a direction process in improvisational systems. Through this document we are going to show our understanding about improvisational agents and coordination as an improvisational direction, in the sense used in the context of improvisational theatre.

It is argued in this thesis that the use of improvisation in deliberative agents creates the conditions for improvising behaviors, in a coherent way, improving their performance as problem solvers, creating a life illusion for interactive agents and contributing for the enhancement of multiagent systems.

Keywords: Improvisational Agents, Deliberative Agents, Improvisational Direction, Distributed Artificial Intelligence, Multi-agent Systems and Coordination.

1 INTRODUÇÃO

Atualmente vários pesquisadores das áreas de Inteligência Artificial (IA) e de Interfaces Homem-Computador (IHC) têm procurado desenvolver sistemas que sejam úteis, eficientes e de fácil utilização. A maioria destes pesquisadores [BAT 94] [KOD 96a] [HAY 97a] [HAY 98a] [HAY 98b] [STO 96][LOY 97] tem apostado que a tecnologia de agentes animados e personificados pode preencher estas características, engajando os usuários na utilização de sistemas. Além das habilidades tradicionais de raciocínio, aprendizagem e processamento da linguagem natural, estes agentes precisam de capacidade de improvisação a fim de apresentarem comportamentos interessantes e coerentes que trazem a ilusão de vida ao agente. Esta ilusão de vida tem se mostrado fundamental para um melhor engajamento dos usuários às tarefas realizadas através de um sistema computacional [LES 97a] [LES 97b] [MUL 98].

O primeiro trabalho em improvisação surgiu com o paradigma da improvisação dirigida criado por Barbara Hayes-Roth no Projeto Teatro Virtual [HAY 94a] [HAY 94b][HAY 95a][HAY 95b]. O paradigma da improvisação dirigida pressupõe a existência de um mundo virtual habitado por agentes atores capazes de interagir com o usuário e improvisar comportamentos. O usuário informa aos agentes instruções, e os agentes seguem as instruções improvisando comportamentos de acordo com sua personalidade, estado de espírito, convenções sociais, histórico, etc. Hayes-Roth [HAY 98a] [HAY 98b] criou uma nova classe de agentes para os agentes improvisacionais. Contudo, esta nova classe de agentes salienta os aspectos artísticos, relacionados à improvisação, e não a relação do agente com os componentes tradicionais da Inteligência Artificial. Entendemos que agentes improvisacionais devem ser vistos como um tipo de agente deliberativo capazes de solucionar problemas de improvisação. Neste sentido, buscamos identificar dentro de uma arquitetura clássica de agentes deliberativos, a arquitetura BDI (*belief-desire-intention*), a possibilidade da inclusão de componentes de improvisação e a partir deste momento, criar uma arquitetura BDI improvisacional e aplicá-la ao caso particular de uma arquitetura de guias virtuais.

A fim de manter a ilusão de vida provocada por um agente individual, é necessário que os agentes organizados em sociedades, com o objetivo de solucionar problemas complexos, não quebrem a ilusão de vida que se tenta projetar. Para manter esta ilusão de vida os agentes precisam agir de forma coerente, ou seja, precisam estar coordenados. Neste sentido, a coordenação é uma das subáreas que tem recebido bastante atenção dos pesquisadores em sistemas multiagentes (SMA). Entendemos que uma das maneiras de se

projetar esta coordenação é através da direção improvisacional, no sentido que este termo é usado no contexto do teatro improvisacional, podendo esta ser centralizada ou descentralizada. A direção, neste caso, é vista como um tipo de coordenação como o planejamento e negociação, mas é um tipo de coordenação especial porque está voltada a agentes que possuem capacidades de improvisar comportamentos durante uma determinada interação. Desta maneira, a improvisação é uma característica chave, estando presente tanto processo de direção quanto nos agentes que serão dirigidos.

Este documento encontra-se organizado em quatorze capítulos. No segundo capítulo são apresentadas as motivações e contribuições esperadas com esta tese. No terceiro capítulo são apresentados os trabalhos relacionados a agentes improvisacionais. No quarto capítulo são descritos os conceitos de teatro improvisacional e agentes improvisacionais e proposta uma nova visão para agentes improvisacionais. No quinto capítulo são detalhadas as funções de um diretor improvisacional. No sexto capítulo são apresentados os tipos de improvisação identificados com base nas fundamentações teóricas descritas nos capítulos quatro e cinco. No sétimo capítulo é apresentado o aspecto da direção improvisacional como coordenação de agentes. No oitavo capítulo são descritos os pressupostos teóricos das arquiteturas de crenças, desejos e intenções (BDI) e alguns exemplos de arquiteturas. No nono capítulo é realizada a identificação dos tipos de improvisação, apresentados no capítulo seis, nas arquiteturas BDI descritas no capítulo oito. No décimo capítulo é apresentada uma proposta de arquitetura BDI improvisacional. No décimo primeiro capítulo é realizada a aplicação da arquitetura BDI improvisacional para uma arquitetura de guias virtuais. No décimo segundo capítulo é apresentado um estudo de caso da arquitetura de guias virtuais para o museu virtual SAGRES. No décimo terceiro capítulo estão as conclusões da tese e no décimo quarto capítulo estão listadas as referências bibliográficas utilizadas para o desenvolvimento desta tese.

2 MOTIVAÇÕES E CONTRIBUIÇÕES ESPERADAS

Muitos trabalhos na área de Inteligência Artificial (IA) têm procurado criar, especificar e desenvolver sistemas compostos por agentes inteligentes. Tradicionalmente, o conceito de inteligência dentro da IA tem sido abordado, de uma maneira geral, como a capacidade de um agente em raciocinar, aprender e se comunicar através de linguagem natural. Estas capacidades têm sido estudadas com a finalidade de fazer com que um agente busque solucionar um problema de maneira completa e correta. A fim de atingir esta correção e completude, os agentes utilizam mecanismos para planejar suas ações com antecedência, tentando prever o que pode acontecer no futuro. Contudo, estes mecanismos têm se mostrado ineficientes, pois necessitam de um amplo conhecimento sobre os fatos futuros, informação que os agentes não possuem e que irá influenciar na sua performance durante a execução. Quando se encontram frente a um problema desconhecido, os agentes gastam muito tempo para replanejar e executar um novo curso de ação.

Neste sentido, Hayes-Roth e Doyle [HAY 98b] salientam que nem sempre as pessoas têm informação ou motivação suficiente para realizar um planejamento completo de suas atividades diárias e, muitas vezes, quando o fazem se encontram em situações inesperadas, que exigem uma resposta rápida e espontânea. Para superar momentos como estes, as pessoas utilizam a sua capacidade de improvisação. Para Hayes-Roth e Doyle em situações complexas e imprevisíveis é melhor improvisar do que planejar.

Compartilhamos desta idéia e entendemos que a improvisação deve ser considerada uma característica de agentes assim como o raciocínio, aprendizagem e processamento da linguagem natural. Em Moraes [MOR 2001a] [MOR 2001b], após realizar a análise de alguns trabalhos em agentes inteligentes personificados, constatamos que a improvisação estava presente, seja de maneira explícita ou implícita, em todos os trabalhos estudados. Isto veio ao encontro da confirmação da nossa hipótese de que a improvisação é uma característica inerente aos agentes.

Embora tenha sido observada a presença da improvisação em agentes personificados, este assunto não tem recebido muita importância. Uma prova disto é a existência de poucos trabalhos enfocando explicitamente os detalhes dos processos de improvisação. Desta maneira, a falta de pesquisas de detalhes em processos improvisação de agentes e a importância que acreditamos ter a improvisação para o desenvolvimento de agentes serviram como motivação para a realização desta tese. Seguindo a idéia dos métodos de resolução de problemas onde um objetivo a ser alcançado é subdividido em vários subobjetivos, a nossa motivação, vista como um objetivo, está subdividida em várias submotivações que são:

- a falta de uma definição clara de improvisação aplicada à computação. A definição de agentes improvisacionais proposta por Hayes-Roth [HAY 94a] [HAY 94b] [HAY 95b] [HAY 95c] apresenta os conceitos de improvisação como base para um novo tipo de agente, mas não aprofunda o relacionamento deste novo tipo de agente com os conceitos tradicionais de agentes inteligentes. No sentido de aprofundar este relacionamento, desenvolvemos alguns trabalhos sobre agentes improvisacionais e sobre uma das mudanças propostas por Hayes-Roth e Doyle de que planejamento se torna improvisação [MOR 2001a] [MOR 2002a] [MOR 2002b]. Contudo, constatamos que, talvez por influência das referências bibliográficas, nosso conceito inicial de agentes improvisacionais nesses trabalhos também enfocava mais os aspectos artísticos do que estávamos querendo mostrar. Iremos retomar nosso conceito, a fim de mostrar mais claramente a relação dos agentes improvisacionais com aspectos tradicionalmente estudados em agentes. Nossa proposta é entender agentes improvisacionais como um tipo de agente deliberativo e não como uma outra categoria separada de agentes como proposto por Hayes-Roth e Doyle [HAY 98b];
- a falta de trabalhos enfocando o aspecto da direção improvisacional e o entendimento do processo de direção como um processo de coordenação de agentes. A maioria dos trabalhos na área, como veremos a seguir na análise dos trabalhos relacionados, considera que a improvisação ocorre somente no agente visto como um ator, ou seja, considera-se a improvisação de comportamentos do agente na sua função de ator. Os trabalhos não examinam outra figura importante no teatro improvisacional, o diretor. Como um diretor teatral é na realidade um coordenador de várias atividades relacionadas à montagem e execução de uma peça de teatro, entendemos que o processo de direção improvisacional é um tipo específico de coordenação de agentes, pois considera aspectos de improvisação. Propomos examinar as funções de um diretor improvisacional e considerá-las em uma arquitetura que possibilite tanto a realização de improvisação por parte de atores quanto a realização de improvisação por parte do diretor, sendo que a improvisação realizada por um diretor é na realidade uma coordenação improvisada de agentes atores e envolve, entre outras coisas, um processo de resolução de problemas.
- a confusão de conceitos na área de coordenação de agentes. Propomos a definição de um espaço de coordenação, esclarecendo os conceitos envolvidos em coordenação para incluir dois aspectos, o da improvisação e o da direção.

As contribuições esperadas com o desenvolvimento deste trabalho são:

- a demonstração de que a improvisação é uma característica básica para agentes inteligentes, assim como raciocínio, aprendizagem e processamento da linguagem natural;
- uma definição clara de improvisação e tipos de improvisação;
- uma definição clara de agentes improvisacionais, fazendo a sua relação com agentes deliberativos baseados em arquiteturas BDI;

- a identificação do uso do conceito de improvisação dentro de arquiteturas BDI consideradas tradicionais;
- a elaboração e especificação uma arquitetura BDI improvisacional;
- a demonstração do uso da improvisação na resolução de problemas como uma característica diferencial de uma arquitetura BDI improvisacional;
- a abordagem ao aspecto de direção de agentes improvisacionais usando como fundamentação teórica a direção no teatro improvisacional. Alguns aspectos da visão de agentes como atores já foram tratado por Hayes-Roth [HAY 94a] [HAY 95d] [HAY 97b] [HAY 99b] [HAY 99c] no projeto Teatro Virtual;
- a elaboração um espaço de coordenação de agentes, incluindo a improvisação e distribuição da coordenação;
- a ilustração como aspectos artísticos da direção podem ser aplicados para direção de agentes computacionais;

3 TRABALHOS RELACIONADOS

Neste capítulo iremos apresentar alguns trabalhos relacionados à improvisação de agentes e discutir a existência, ou não, de aspectos de direção improvisacional nestes trabalhos.

3.1 Projeto Teatro Virtual

O Teatro Virtual [HAY 94a] [HAY 94b] [HAY 95a] [HAY 97a] [HAY 97b] é um projeto desenvolvido no Departamento de Ciência da Computação da Universidade de Stanford sob a coordenação da pesquisadora Barbara Hayes-Roth. Este projeto tem como objetivo proporcionar um ambiente multimídia no qual os usuários podem executar papéis criativos associados à produção e execução de peças e histórias numa companhia de teatro improvisacional. Estes papéis são: produtor, diretor e ator. Através destes papéis, os usuários podem interagir com agentes atores, informando aos mesmos instruções abstratas que determinam a estrutura narrativa da história. Os agentes atores são agentes animados que improvisam comportamentos físicos e verbais que seguem a narrativa e atendem a outros critérios de execução, tais como, qualidades individuais e convenções sociais. Deste modo, caso uma mesma história seja executada sucessivamente, os agentes poderão improvisá-la de diferentes maneiras.

A arquitetura dos agentes atores utilizada no projeto Teatro Virtual pode ser visualizada na figura 3.1 [HAY 95a]. Um agente possui um corpo, uma mente e uma interface mente-corpo. O corpo do agente é um programa de computador que controla a manifestação gráfica do agente no mundo virtual. O corpo possui sensores e atuadores que percebem eventos e executam comportamentos. A mente do agente é um programa de computador responsável por executar três funções: integrar as entradas percebidas com conhecimentos e inferências para atualizar a representação da situação dinâmica do agente, instanciar e decidir quando executar determinados comportamentos e executar todo o processamento necessário entre a escolha de uma nova situação e execução dos comportamentos relacionados a ela. A interface mente-corpo realiza o intermédio entre a interação mente e corpo através de um *loop* de controle que ocorre em tempo real. Este intermédio é realizado através do mapeamento de informações sensoriais para percepções significativas (que são enviadas para a mente) e da tradução de comandos, fornecidos pela mente, em seqüência de comandos executáveis relacionados aos atuadores.

A mente e a interface mente-corpo são implementadas como controladores assíncronos para comportamentos cognitivos e físicos. Cada controlador implementa um modelo de controle dinâmico, permitindo continuamente ao agente notificar possíveis

comportamentos, descrever seu comportamento em planos de controle abstratos, gerar e modificar seus planos de controle em tempo de execução [HAY 95b]. A cada instante de tempo o agente seleciona e executa comportamentos que combinam com o seu plano de controle corrente. Isto é realizado através de interações entre o controlador cognitivo (mente) e o controlador físico (interface mente-corpo). O controlador cognitivo do agente recebe entradas do seu controlador físico, constrói um modelo envolvendo a sua situação e a sua interação com outros personagens, planeja seus comportamentos físicos e verbais e envia planos de controle para o seu controlador físico.

O controlador físico do agente interpreta e filtra as entradas da sua interface com o usuário através dos sensores do mundo virtual e envia estas percepções ao seu controlador cognitivo. Ele recebe os planos de controle do controlador cognitivo e desempenha estes planos no seu comportamento físico, enviando as suas saídas para o mundo virtual e para a interface com o usuário.

O controlador cognitivo do agente interage em um ciclo de três passos:

- gerenciador da agenda, responsável por habilitar os comportamentos possíveis para a situação atual. Para realizar esta tarefa ele recebe informações sobre os eventos que ocorreram no mundo real e no mundo virtual. Ele também utiliza informações relativas a quais comportamentos foram executados previamente. Estas informações são obtidas através da memória;
- escalonador, responsável por escolher, dentre os comportamentos possíveis gerados pela agenda, qual o melhor comportamento para satisfazer o plano de controle corrente;
- executor, responsável por executar o comportamento escolhido.

Os comportamentos de um agente são instâncias do seu repertório de classes cognitivas e físicas e são organizados em hierarquias de classes. Cada comportamento possui um conjunto de condições que podem ser satisfeitas pela ocorrência de determinadas classes de eventos. Quando as condições para um determinado comportamento são satisfeitas, o gerenciador da agenda habilita o comportamento mais apropriado para a situação corrente.

Para assumir um dos papéis (produtor, diretor ou ator) o usuário opera em um determinado modo de interação. Os modos possíveis são: bonecos animados, atores animados, modo do diretor, modo de construção colaborativa e modo interativo de refinamento do *script*. Dependendo do modo de interação que está sendo executado, cada agente, visualizado como um personagem animado, aceita orientações de uma ou mais fontes, em tempo real ou no decorrer de uma execução. As direções são representadas como restrições, mas não especificam completamente o comportamento do agente. Cada agente deve construir o seu próprio curso de comportamento, seguindo as direções, interagindo com outros agentes e conhecendo o ambiente de improvisação ao longo da execução.

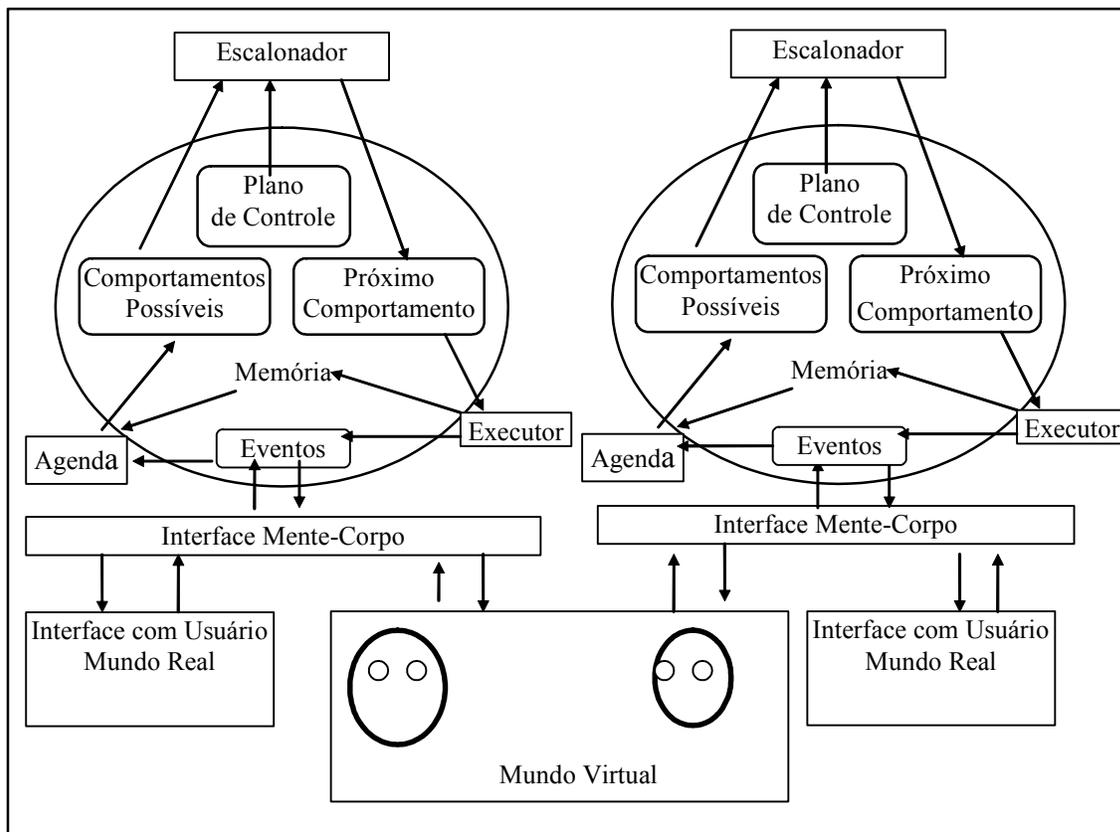


Figura 3.1: Arquitetura dos Agentes

No modo de bonecos animados, os usuários interagem com uma interface gráfica e utilizam menus de comportamento atual para dirigir as improvisações dos agentes interativamente. A figura 3.2 mostra a interface gráfica do usuário para o modo de bonecos animados. No modo de atores animados, os usuários dirigem os agentes fornecendo a eles *scripts* de comportamentos abstratos para histórias pré-concebidas, que os seus personagens executam de maneira autônoma. No modo de diretor, os usuários preparam *scripts* de comportamentos abstratos para os atores animados, mas continuam a dirigi-los ocasionalmente durante a execução através do menu de comportamentos abstratos mostrados na figura 3.2. No modo de construção colaborativa, os usuários usam o modo de bonecos animados para improvisar histórias. Quando eles observam algo de que gostam, eles armazenam o segmento de *script* de comportamento correspondente no seu repertório para que ele possa ser utilizado posteriormente. No modo interativo de refinamento do *script*, os usuários constroem *scripts* de comportamentos abstratos para os atores animados da seguinte maneira, quando um ator improvisa algo, o usuário cria um *script* de comportamentos instanciados correspondente a esta improvisação.

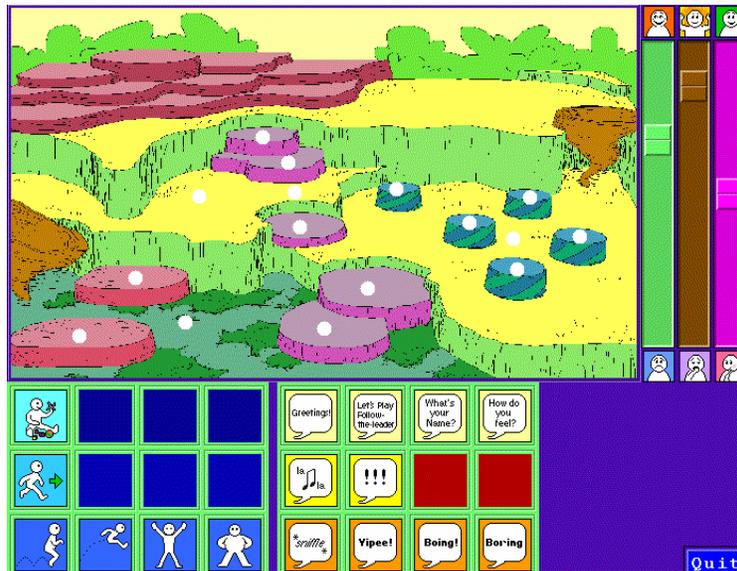


Figura 3.2: Interface do Modo de Bonecos Animados disponível em www-ksl.stanford.edu/cait/demos/woggles.html

Podemos observar que todos os modos de interação proporcionam a direção de agentes atores. Alguns modos proporcionam a direção direta de comportamentos através da manipulação do menu de comportamentos, e outros modos proporcionam a direção indireta de comportamentos através da comunicação de *scripts* de comportamento para os agentes atores. Contudo, estas direções são sempre realizadas por usuários. O usuário efetua o papel de diretor na construção da peça interativa. Não existe a figura de um agente diretor ou de um processo automatizado para realização da direção dos atores.

3.2 Empresa Extempo Systems Inc.

A empresa Extempo Systems Inc. (<http://www.extempo.com>) foi fundada em 1996 por Barbara Hayes-Roth, diretora do projeto Teatro Virtual do Departamento de Ciência da Computação da Universidade de Stanford. Hayes-Roth utilizou suas pesquisas fundamentais em agentes para criar sua empresa, que desenvolve personagens para aplicação na Web ou em outros ambientes interativos. Os personagens desenvolvidos possuem personalidades distintas e executam diferentes papéis em várias aplicações. Os personagens são chamados de *imp characters* (personagens improvisacionais), porque eles improvisam seus próprios comportamentos interativos, incluindo ações, gestos e conversação em tempo real [HAY 99a].

Devido à influência dos trabalhos anteriormente realizados no projeto Teatro Virtual, a arquitetura dos personagens (figura 3.3) desenvolvidos pela Extempo também possui os componentes de mente, corpo e interface mente-corpo. O servidor de mente (*imp mind server*) roda as mentes do personagem que se comunicam através de uma interface mente-corpo para controlar seu próprio corpo que está rodando no computador do usuário. A interface de programação do servidor permite a integração com banco de dados, arquivos de perfis de usuários ou sistemas para oferta de produtos [HAY 99b].

A interface mente-corpo permite diferentes apresentações para o corpo (tais como animações, voz, Microsoft Agent ou Extempo LiveComics) e diferentes modos de interação do usuário com o personagem (tais como voz, texto e botões).

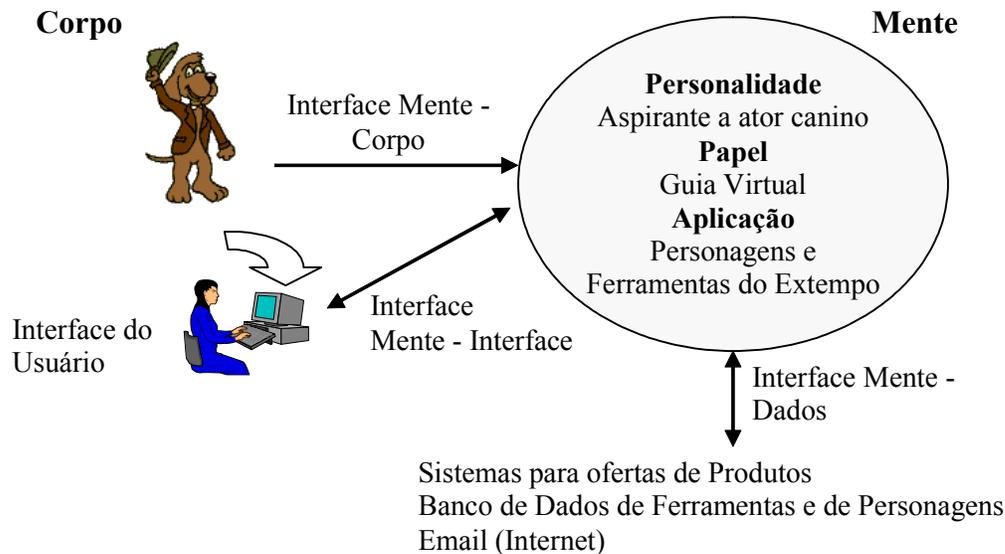


Figura 3.3: Arquitetura baseada em componentes para personagens interativos

A mente do personagem é modelada usando a ferramenta ICDK (*Imp Character Development Kit*). Esta ferramenta possibilita a criação, teste e avaliação de personagens [HAY 2001] [EXT 2000]. A tecnologia relacionada à mente do personagem fornece certas capacidades cognitivas básicas como: gerenciamento da interação, estados de espírito dinâmicos, entendimento da linguagem natural, geração da linguagem natural, gerenciamento do diálogo, seleção e coordenação de ações e conversação. A mente de um personagem é composta de três componentes: uma personalidade única e persistente, uma alternativa de papel e uma aplicação específica (figura 3.3). Estes três componentes correspondem aos três construtores para agentes (personagem, papel e enredo) que serão descritos na seção 4.2.1.

A personalidade do personagem descreve todas as características de um personagem individual. As características representadas são: histórico do personagem; aparência; maneira de se mover, de se movimentar e de falar; voz; conhecimento; opiniões e dinâmicas sociais e emocionais. Um papel é uma coleção de conhecimentos e comportamentos que um personagem precisa para executar um determinado trabalho como, por exemplo, guia virtual, vendedor ou *bartender*. Vários personagens podem executar um mesmo papel, contudo cada um possui uma maneira de interpretar o papel que lhe foi destinado. Isto é possível porque os papéis são personalizados para cada personagem. Uma aplicação é uma coleção de conhecimento e comportamento que o personagem precisa para executar um determinado papel em um contexto particular. É necessário que o personagem possua informações sobre o *site* ou produto que ele deve apresentar, a fim de poder auxiliar o usuário na navegação e comércio eletrônico [HAY 99b].

Para criar seus personagens os autores devem selecionar ou modificar uma personalidade, um papel e uma aplicação. Para cada um destes componentes, ICDK fornece

templates para especificar os comportamentos e conhecimentos específicos de um personagem, incluindo elementos para entendimento da linguagem natural, geração da linguagem natural, dinâmica dos estados de espírito e coordenação de diálogos e ações. Os *templates* oferecem elementos pré-definidos que podem ser adaptados para um determinado personagem. Os autores podem modificar os *templates*, a fim de incluir variabilidade nos comportamentos do personagem. Esta variabilidade é conseguida através da especificação implícita ou explícita de várias alternativas de diálogo ou de comportamento para uma dada situação.

Para manipular a ferramenta de autoria dos personagens (ICDK), os autores não precisam ter conhecimento de programação. Eles simplesmente selecionam e modificam *templates*, e a ferramenta automaticamente gera o personagem, incluindo entradas no banco de dados que contém as informações relativas ao personagem. A qualquer momento o autor pode compilar e executar o personagem a fim de visualizar como o mesmo está ficando.

A figura 3.4 mostra a interface da ferramenta ICDK quando o usuário está criando os tópicos relacionados à interação do personagem com o usuário.

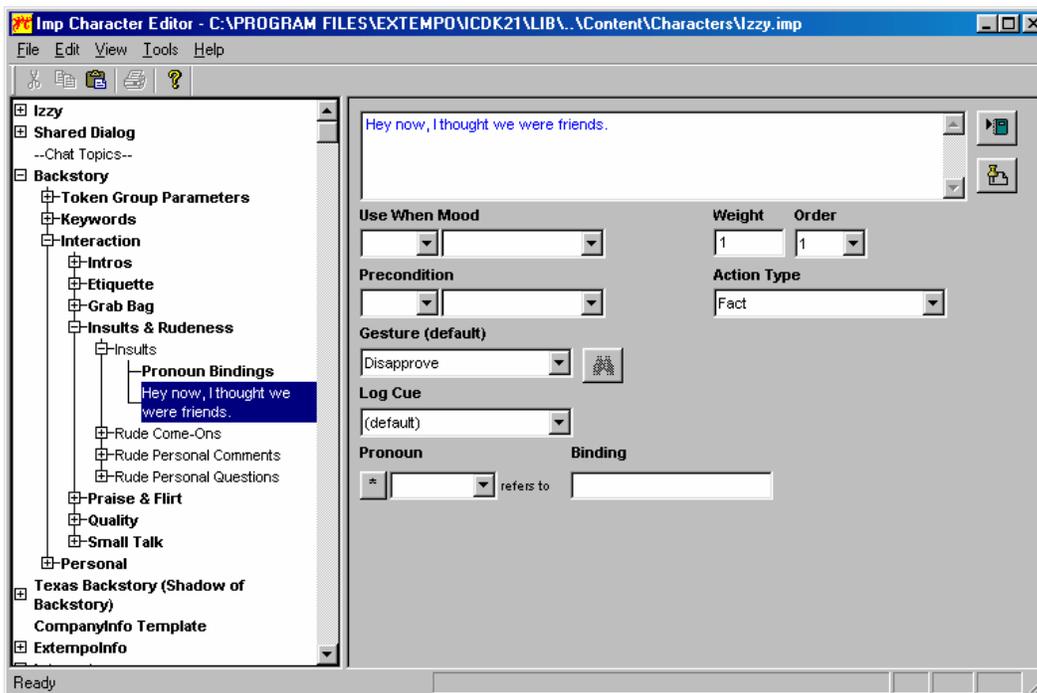


Figura 3.4: Interface da ferramenta ICDK para criação de tópicos de interação com o usuário

A partir das explicações anteriores podemos observar que existem duas possibilidades de direção dos personagens. Uma direção direta das ações e comportamentos dos personagens é realizada pelo autor no momento em que o mesmo faz a construção do personagem usando a ferramenta ICDK. Uma direção indireta das ações e comportamentos é realizada pelo usuário quando o mesmo interage com o personagem. Através da interação o usuário pode, entre outras coisas, elogiar ou insultar o personagem o que irá refletir

diretamente nas suas ações e comportamentos. A maneira como o personagem reage à interação é dirigida pelo autor. Contudo, se o usuário ao interagir com o personagem percebe que ele se mostra mais amigável e lhe fornece mais ajuda quando é elogiado, o usuário pode dirigir indiretamente o personagem, por exemplo através de elogios, para que ele aja de determinada maneira.

Assim como no projeto Teatro Virtual, a direção dos personagens na ferramenta da Extempo somente é realizada por agentes humanos. Contudo, no Teatro Virtual somente o usuário dirigia os personagens e na ferramenta da Extempo não somente o usuário, mas também o autor do personagem pode realizar uma direção. Constatamos que, embora a direção dos personagens possa ser realizada por duas fontes distintas, nenhuma delas é um processo automatizado de direção, não existe a figura de um personagem ou agente diretor que gerencia a execução dos seus personagens ou atores. Outro fator a ser destacado é que o personagem atua de maneira individual, não interagindo com outros personagens, apenas com o usuário, que também é um único indivíduo.

3.3 Projeto Oz

O projeto Oz é um projeto da Escola de Ciência da Computação da Universidade Carnegie Mellon que tem como objetivo desenvolver tecnologias relacionadas à construção e manipulação de mundos dramáticos virtuais. O projeto permite que os usuários interajam com micromundos para criar e apresentar dramas interativamente. Um drama interativo é definido como a união de personagem (agente), história e apresentação.

A arquitetura geral do projeto Oz pode ser visualizada na figura 3.5.

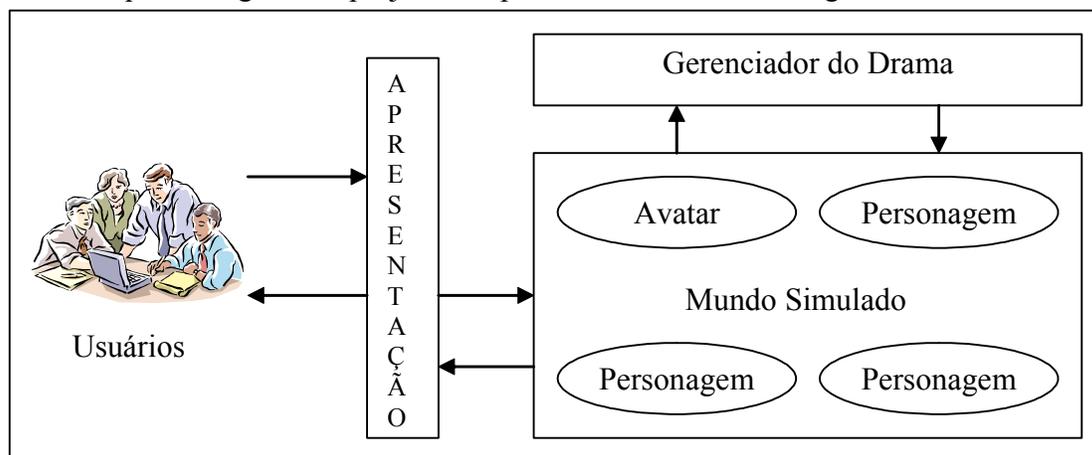


Figura 3.5: Arquitetura Geral do Projeto OZ

A figura 3.5 mostra que o mundo simulado possui personagens e um avatar. Os personagens são agentes computacionais que exibem personalidades, emoções, comportamentos sociais e objetivos. O avatar é um agente dirigido pelo usuário e o representa dentro do mundo simulado. O usuário interage com o mundo através do seu avatar e de uma apresentação. O gerenciador do drama observa todos os acontecimentos do mundo simulado e tenta guiar a experiência do usuário, a fim de fazer uma determinada história se desenvolver. Esta tarefa pode envolver mudanças no modelo do mundo físico como a introdução e retirada de personagens para o direcionamento de um curso de ação e

desenvolvimento de uma história. A figura 3.6 mostra em detalhe a arquitetura geral do projeto Oz, previamente apresentada na figura 3.5 [LOY 91] [REI 92].

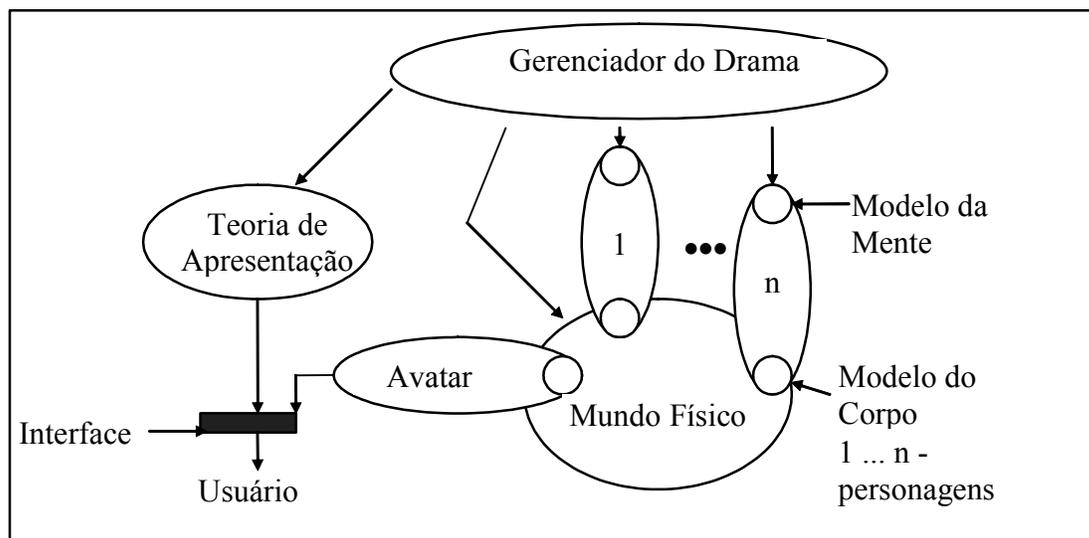


Figura 3.6: Arquitetura detalhada do projeto Oz [KEL 91]

Através da figura 3.6 se pode observar que o modelo do corpo de cada personagem e do avatar estão no mundo físico. Logo, o mundo físico contém a representação visual dos componentes presentes no ambiente. Fora do mundo físico, o modelo de mente controla todas as ações dos personagens. As ações executadas pelo avatar são controladas pelo usuário. Informações sensoriais são transmitidas do mundo físico para o avatar através de uma interface controlada por uma teoria de apresentação. A teoria de apresentação continuamente ajusta o estilo de apresentação, a fim de acompanhar as mudanças no conteúdo dramático que ocorrem durante a interação com o ambiente. O gerenciador de drama influencia a mente dos personagens, o mundo físico e a apresentação da teoria, planejando e controlando o fluxo de eventos que ocorre no ambiente [KEL 91] [BAT 91b].

Uma das principais características dos mundos virtuais é fazer com que as pessoas sejam capazes de imaginar que o mundo simulado é o mundo real. Segundo Bates [BAT 91b], para fornecer esta ilusão da realidade é necessário que os agentes tenham capacidades amplas. Com este objetivo foi projetada uma arquitetura de agentes, chamada Tok, que inclui objetivos, comportamentos reativos, estados emocionais e os seus efeitos no comportamento, alguma capacidade de uso da linguagem natural e conhecimento dos agentes (dele mesmo e dos outros) e do mundo físico.

O Tok possui três componentes principais: uma máquina reativa baseada em objetivos, chamada Hap, um componente de emoção chamado Em, e um sistema de linguagem natural chamado Glinda. A figura 3.7 mostra a arquitetura do Tok.

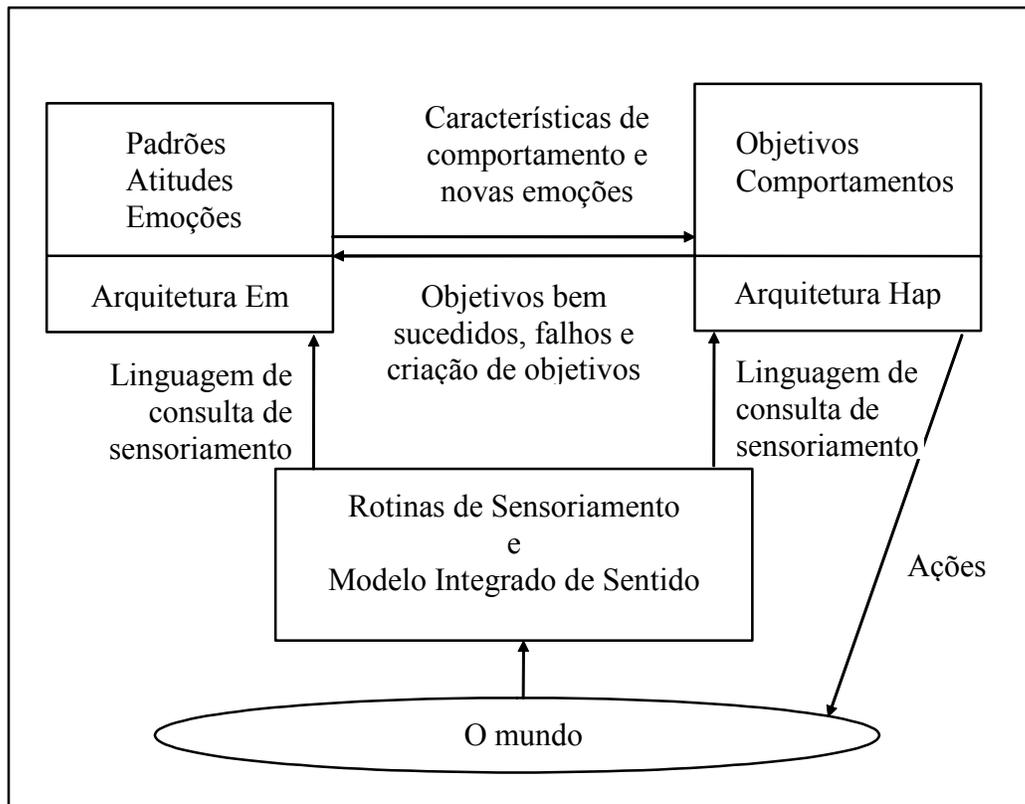


Figura 3.7: Arquitetura do Tok [BAT 92a]

O Hap é responsável por controlar os personagens do mundo Oz. O mundo Oz proporciona uma simulação orientada a objetos, onde pode ocorrer tanto a interação entre vários personagens humanos e de computador quanto a criação de novas histórias interativas. O mundo se conecta aos personagens através de canais de percepção-ação. Quando um personagem quer executar uma ação, ele envia um frame de ação para o seu corpo no mundo, onde a ação é executada. A ação executada altera o mundo propagando informação sensorial e retornando uma indicação de sucesso ou falha. Os personagens sentem os objetos e eventos do mundo através de dados sensoriais, que são uma coleção de pares atributos-valores. Exemplos de atributos são tamanho, cor, falas de texto, etc[LOY 91].

A fim de satisfazer estas características, o Hap é um planejador reativo dirigido a objetivos. Ele continuamente escolhe a próxima ação do agente baseado na percepção, objetivo corrente, estado emocional e aspectos do estado interno. Os objetivos do Hap são representados através de um nome atômico e um conjunto de parâmetros que são instanciados quando o objetivo se torna ativo. Os objetivos não caracterizam necessariamente estados do mundo que devem ser atingidos, e o Hap não realiza um planejamento explícito. Ao contrário, conjuntos de ações ou planos, são escolhidos de uma biblioteca de planos que pode conter um ou mais planos para cada objetivo. Estes planos são coleções de subobjetivos e ações primitivas. As ações primitivas são de dois tipos, físicas e mentais. Os subobjetivos são construídos pelo autor, as ações físicas são

fornecidas pelo domínio e as ações mentais são construídas através de uma linguagem (Lisp ou C) e incluem ações tais como armazenamento de fatos, execução de cálculos [BAT 92a][LOY 93][LOY 97].

O Hap suporta a produção de comportamentos a partir de ações dirigidas aos objetivos e permite codificar tarefas cognitivas. Os comportamentos são escritos para descrever as atividades que o agente pode se engajar. Cada comportamento é escrito para um objetivo particular e é incorporado ao agente quando um objetivo com aquele nome é atingido [LOY 97].

Junto com a reatividade, as emoções são um aspecto fundamental para atingir a credibilidade, ou seja a ilusão de vida. A geração de emoções no componente Em da arquitetura Tok é baseada nas idéias de Ortony, Clore e Collins. Desta maneira, Em desenvolve emoções de uma base cognitiva, onde eventos externos são comparados com objetivos, ações são comparadas com padrões e objetos são comparados com atitudes [BAT 92a][REI 92]. Eventos são coisas que acontecem, incluindo as ações dos agentes e eles são julgados como agradáveis ou desagradáveis de acordo com os objetivos dos agentes. As ações dos agentes podem ser aprovadas ou desaprovadas de acordo com um conjunto de padrões. Os padrões representam tanto crenças morais, certo e errado, quanto crenças pessoais quanto ao nível de execução. Os agentes podem gostar ou não de manipular determinados objetos (incluindo outros agentes) devido a sua atitude. A atitude do agente representa os seus gostos e preferências. Em possui cerca de 20 tipos de emoções. Algumas destas emoções são: alegria e angústia, que ocorrem, respectivamente, quando os objetivos dos agentes são bem sucedidos ou falham, esperança que ocorre quando Em acredita que existe alguma chance do seu objetivo ser alcançado e medo que ocorre quando existe alguma chance do objetivo não ser alcançado [REI 92].

Estes dois componentes da arquitetura Tok, Hap e Em, têm como tarefa produzir uma série de ações que façam com que os personagens pareçam ter vida, isto é sejam creíveis.

Foram desenvolvidos vários trabalhos para tratar todos os tópicos envolvidos no projeto. O trabalho desenvolvido por Loyall [LOY 97] se concentrou na especificação e definição da construção de personalidades para agentes creíveis, o que resultou na máquina reativa Hap. O trabalho de Reilly [REI 96] se concentrou na especificação e desenvolvimento de aspectos emocionais e sociais de agentes creíveis, o que resultou no componente de emoção Em. O trabalho realizado por Smith e Bates [SMI 89], por Kantrowitz [KAN 90] e por Kantrowitz e Bates [KAN 92] se concentrou no desenvolvimento da máquina de entendimento da linguagem natural Glinda. O trabalho desenvolvido por Weyhrauch [WEY 97] se concentrou na construção do sistema que controla o fluxo da história interativa para criar uma experiência dramática para o usuário. O trabalho desenvolvido por Mateas [MAT 2002] apresenta a arquitetura *Facade* que tem como objetivo integrar personagem, história e processamento de linguagem natural em um sistema completo. Dentre estes trabalhos o que mais se aproxima de aspectos relacionados à direção é o de Weyhrauch, que trabalha com o componente relacionado ao gerenciador do drama. Neste sentido, nos próximos parágrafos serão apresentadas algumas considerações em relação aos estudos realizados por Weyhrauch.

Weyhrauch [WEY 97] desenvolveu uma arquitetura chamada Moe que tem como objetivo fornecer um acompanhamento da experiência dramática. Os dois componentes principais do processo de tomada de decisão do Moe são: busca adversária e avaliação.

Para guiar a experiência do usuário em cada momento, Moe possui os chamados movimentos Moe (MOE MOVES). Por exemplo, um movimento Moe pode trazer um novo personagem para a cena, dando a este personagem uma personalidade forte que pode afetar o rumo do drama. Através dos movimentos Moe, a arquitetura pode guiar a experiência do usuário para o desfecho preparado pelo artista que construiu o drama interativo. Dependendo da experiência em que o usuário se encontra no momento corrente, pode ser apropriado para Moe realizar um acompanhamento dramático. Neste ponto, Moe deve decidir qual movimento deve ser realizado (se houver algum). Para tanto, Moe usa busca adversária [WEY 97].

A busca adversária se assemelha a uma busca em árvore, onde Moe é um jogador e o usuário é outro jogador. No nível mais abstrato, o gerenciador do drama pode realizar uma série de movimentos. No próximo nível, para cada movimento Moe que o gerenciador do drama tomou, o usuário pode realizar determinadas respostas, cada uma delas representando alguma ação significativa que o usuário tomou. Estas ações significativas são chamadas de movimentos do usuário. No próximo nível da busca, Moe responde a cada movimento do usuário. Este processo vai se repetindo até que a busca atinja o fim da experiência.

Cada caminho da árvore corresponde a uma experiência futura possível. Uma função de avaliação atribui um valor para cada futuro possível, e o mecanismo de busca propaga o valor até chegar à raiz da árvore, a fim de escolher o movimento Moe que maximiza o valor esperado para a experiência. A experiência atual é representada por um conjunto de movimentos Moe e de movimentos do usuário, que são armazenados em um histórico.

A função de avaliação trabalha examinando cada um dos movimentos na experiência completa (histórico mais um futuro possível) de acordo com uma variedade de critérios dramáticos. Os objetivos destes critérios são fazer com que a experiência seja dirigida para o final esperado pelo artista mas sem tirar o sentimento de liberdade de ação do usuário.

A arquitetura Moe proposta por Weyhrauch para dar suporte ao gerenciador do drama, apresenta uma abordagem automatizada para o acompanhamento de experiências virtuais. Como o próprio nome sugere, estas experiências ocorrem dentro de mundos virtuais onde o usuário é representado por um avatar. Embora o usuário sinta ter liberdade total para realizar ações com o seu avatar, as ações que o usuário pode realizar são às vezes dirigidas pelo gerenciador do drama. As ações serão dirigidas quando o usuário estiver se afastando do rumo predeterminado pelo autor da experiência interativa. O objetivo do gerenciador do drama é dirigir a experiência do usuário, mas sem que ele tenha conhecimento de que isto está sendo realizado. Para realizar este tipo de direção, o gerenciador do drama utiliza busca adversária e uma função de avaliação. Na execução da busca adversária, o gerenciador utiliza conhecimento dos movimentos Moe e dos movimentos do usuário. O autor prevê os movimentos do usuário e os organiza em um grafo dirigido acíclico. O gerenciador utiliza esta informação para determinar quando deve dirigir a experiência do usuário.

Embora o gerenciador do drama seja um processo automatizado de direção, o tipo de direção realizado pelo gerenciador é diferente do tipo de direção proposto neste trabalho. O gerenciador do drama se baseia em um procedimento de busca chamado de busca adversária, que pode ser entendido como um tipo de planejamento. Mais adiante veremos algumas limitações do planejamento e como a questão do planejamento irá ser abordada neste trabalho. Além desta questão, existem outras diferenças entre a direção realizada pelo gerenciador do drama e a direção que pretendemos realizar. Na seção 3.5 discutiremos sobre estas diferenças.

3.4 Projeto Improv NYU

Improv é um projeto desenvolvido no laboratório de pesquisa em mídia da Universidade de Nova Iorque. O objetivo do projeto é criar um conjunto de ferramentas que tornam possível desenvolver aplicações envolvendo agentes animados que se comportam de maneira semelhante aos humanos, incluindo sentimentos imprevisíveis que permanecem consistentes com a personalidade e objetivos definidos para o personagem [PER 96a] [PER 96b].

Improv é composto de dois subsistemas. O primeiro subsistema é uma máquina de animação e o segundo subsistema é uma máquina de comportamento que permite aos autores criarem regras sofisticadas para especificar como os atores se comunicam, se adaptam e tomam decisões. A combinação destes dois subsistemas fornece um conjunto de ferramentas integradas para construção de mentes e corpos de atores interativos. O sistema utiliza uma linguagem de *script* baseada na língua inglesa, possibilitando aos especialistas, que não são programadores, uma facilidade para criar aplicações interativas.

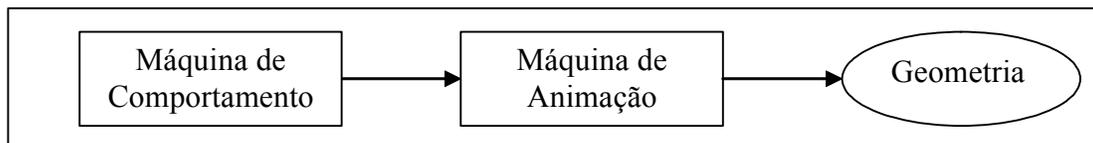


Figura 3.8: Arquitetura básica para um ator no projeto Improv

O modelo de comportamento utilizado pelo Improv é semelhante ao proposto por Blumberg [BLU 95] que consiste de uma geometria (figura) que é manipulada em tempo real. A arquitetura do Improv é composta por uma máquina de animação, que utiliza descrição de ações atômicas animadas, tal como caminhar ou abanar, para manipular a geometria, e uma máquina de comportamento que é responsável pelas capacidades de alto nível, tal como ir a uma loja, conversar com outros atores e tomar decisões sobre qual animação executar. A máquina de comportamento também é responsável por manter o modelo interno do ator, representando vários aspectos como estados de espírito, objetivos e personalidade. Desta maneira, a máquina de comportamento constitui a mente do ator [PER 96b].

Os movimentos e comportamentos dos atores são computados em tempo real através do ciclo de interação que ocorre entre as máquinas de animação e de comportamento.

A máquina de animação é responsável por fornecer ferramentas para geração e tratamento interativo de gestos e movimentos realísticos. Os atores são capazes de se mover de maneira natural, pois seus movimentos são representados em camadas que cobrem diferentes estados de espírito e personalidades. Em outras palavras, a máquina de animação controla o corpo do ator.

A mente de ator é controlada através da máquina de comportamento. A máquina de comportamento permite a um autor especificar como o ator deve agir através da descrição e configuração de três componentes principais: *scripts* de comportamento, regras de comportamento e personalidade do ator.

Os *scripts* permitem criar conjunto de ações e comportamentos complexos. Os *scripts* podem ser utilizados para ativar outros *scripts* e/ou ações. Os *scripts* fornecem aos projetistas de mundos virtuais uma maneira fácil de se construir uma hierarquia de comportamentos, que vão desde decisões de baixo nível como que tipos de gestos e ações devem ser executados até comportamentos de alto nível que descrevem os objetivos do ator e os tipos de atividades sociais que ele pode realizar. Desta maneira, os objetivos e atividades podem ser especificados através de componentes de comportamento, que por sua vez podem ser especificados em atividades físicas individuais.

Para se comportar de maneira semelhante aos humanos os atores animados devem apresentar comportamento credível. Isto implica que os atores animados devam executar as ações levando em conta as características do ambiente, a interação com o usuário e suas propriedades internas. Os autores podem criar regras de decisão para obter informações sobre um ator e seu ambiente e utilizá-las para determinar as tendências do ator sobre determinadas escolhas ao invés de outras. A figura 3.9 mostra um exemplo de um *script* com regras de decisão.

Além de criar *scripts* e regras de decisão que influenciam o comportamento do ator durante a execução de um determinado *script*, os autores realizam a definição da personalidade do ator. Através da definição da personalidade os autores especificam o temperamento, estados de espírito e relacionamento do ator com outros atores e objetos. A figura 3.10 mostra um exemplo de definição de personalidade.

SCRIPT “Vá para a Loja”	
passos	
1.	Sair de casa
2.	escolha entre (Caminhar, Pegar ônibus, Pegar táxi, Ir de bicicleta) usando critério “melhor maneira de ir para lá”
3.	Entrar na loja
4.	...
CRITÉRIO “maneiras de se chegar lá”	
critério	importância
1.minha energia comparada com o preparo físico	.7
2.minha saúde comparada com o custo	.3

Figura 3.9: Exemplo de script associado a regras de decisão

O Improv possibilita a descrição de vários tipos de comportamentos, que podem ser executados através de seqüências lineares de *scripts* e ações ou através de *scripts* que incorporam processos de tomada de decisão baseados na personalidade do ator. Desta maneira, um mesmo ator pode executar *scripts* lineares e/ou *scripts* mais complexos envolvendo processo de tomada de decisão, realizado através das regras de decisão. Para que isto ocorra os autores podem especificar de maneira mista os comportamentos e níveis de controle da autonomia do agente, criando assim vários efeitos possíveis para os atores.

ATOR "Gregor"	
atributo	valor
Inteligência	.99
Amabilidade	.5
Força	.33
Coordenação	.3
Simpatia relacionada a Otto	.15
Simpatia relacionada à Gabriela	.9
Conhecimento sobre Grécia Antiga	.75
Habilidade de falar em público	.9
Habilidade de dirigir carros	.25

Figura 3.10: Exemplo da definição da personalidade do ator no momento de sua criação

O projeto Improv permite dar a um autor o mesmo controle que ele possui sobre um ator individual para um grupo de atores. Desta maneira, o autor exerce o papel de um diretor que pode dirigir o drama através de regras de comportamentos previamente escritas. O autor pode enxergar o conjunto de atores como um elenco que pode ser coordenado da mesma maneira que um ator individual [PER 96b]. Para que isto seja possível, os atores podem modificar as propriedades de outros atores da mesma maneira que eles modificam suas próprias propriedades. Assim, quando um ator realizar uma ação ele pode modificar as propriedades dos outros atores de maneira que os mesmos reajam imediatamente à ação. Os atores se comunicam através de um quadro negro compartilhado, como mostra a figura 3.11.

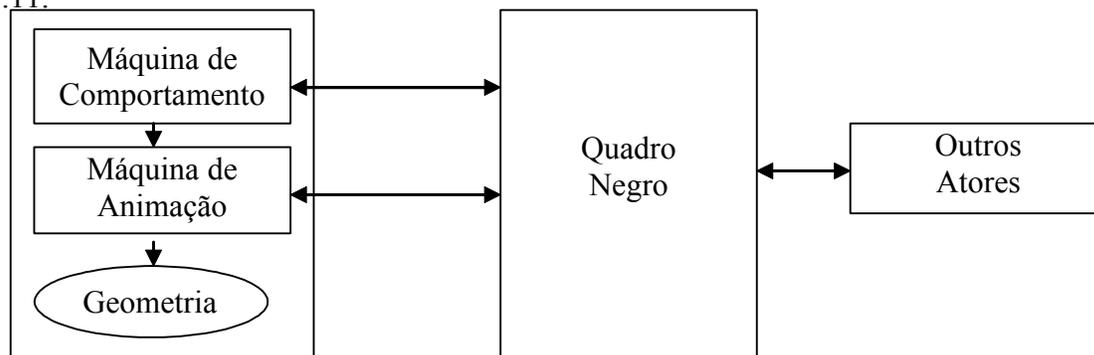


Figura 3.11: Comunicação entre agentes no Improv

Considerando esta breve apresentação do projeto Improv, podemos observar que existem duas formas de direção: uma direção individual e uma direção coletiva de atores. No primeiro caso, a direção é realizada através da autoria de *scripts*, regras de decisão e definições de personalidade para cada ator. No segundo caso, a direção é realizada através de regras de decisões que descrevem comportamento social do ator em relação aos outros atores. Contudo, em ambos os casos quem realiza a direção é o autor da aplicação, não existindo um processo de direção para realização da autoria das regras envolvidas na direção individual e coletiva dos atores.

3.5 Considerações sobre o Capítulo e Trabalhos Anteriores em Direção Improvisada

Em todos os projetos apresentados existe algum tipo de direção de agentes, mas esta direção na maioria das vezes está vinculada a um agente humano, seja ele um autor de aplicação ou um projetista de software. Somente no projeto Oz existe a presença de um processo automatizado para realizar um tipo de direção, o gerenciador do drama.

Como mencionado anteriormente, o gerenciador do drama realiza a direção da experiência do usuário em um mundo virtual. Esta direção é executada através de uma análise que verifica se o usuário está experimentando a experiência interativa proposta por um artista. O objetivo do gerenciador é dirigir a ação do ambiente virtual no sentido de atingir esta experiência. O tipo de direção realizada pelo gerenciador do drama é diferente do tipo de direção que estamos propondo. A nossa proposta de processo de direção abrange aspectos diferentes, visto que vamos tomar como base o processo de direção realizado no teatro improvisacional e este processo envolve: a construção de *scripts* de comportamento para agentes atores, considerando suas personalidades e estados de espírito; o gerenciamento de agentes atores; a solução de problemas que podem surgir durante uma interação e a avaliação tanto da performance dos agentes atores quanto do processo de direção propriamente dito.

A idéia inicial de criar um processo automatizado para realizar a direção improvisada de agentes surgiu durante a realização de meu trabalho de mestrado. Na minha dissertação desenvolvi uma arquitetura multiagente composta de um agente diretor e de agentes atores. O diretor tinha como objetivo dirigir e gerenciar um grupo de atores que realizavam o acompanhamento de visitantes ao Museu Virtual SAGRES. Para realizar estas tarefas o diretor era responsável por construir *scripts* de execução de comportamentos para agentes atores. Esta construção era realizada através do relacionamento de comportamentos verbais e físicos às ações dos agentes atores. O conjunto de ações dos agentes era especificado pelo projetista do sistema, e o relacionamento de comportamentos a ações se baseava em um conjunto de comportamentos possíveis dos agentes atores. Após construir o *script* de execução de comportamentos o diretor relacionava este *script* a um ator que executava os comportamentos apropriados para o usuário [MOR 99a] [MOR 99b] [MOR 99c].

Este trabalho se constituiu em uma primeira abordagem para a questão da implementação de um processo de direção improvisacional realizado por um agente e serviu para salientar a importância do diretor, mais especificamente da direção, em uma

atuação improvisada. Contudo, o processo de direção foi considerado de uma maneira simplificada não enfocando todos os aspectos relacionados à direção improvisacional.

A fim de aperfeiçoar o processo de direção automatizada e improvisada de agentes improvisacionais, ou seja, de agentes que têm a capacidade de improvisar comportamentos durante a execução de uma determinada aplicação, foi necessário um aprofundamento dos estudos dos fundamentos teóricos relacionados ao Teatro Improvisacional. Alguns resultados destes trabalhos podem ser encontrados em [COS 2000] [MOR 2001a] [MOR 2001b]. Na próxima seção iremos apresentar alguns aspectos do Teatro Improvisacional necessários para o embasamento desta tese.

4 TEATRO IMPROVISACIONAL COMO FUNDAMENTO PARA AGENTES IMPROVISACIONAIS

Um dos primeiros exemplos da improvisação no teatro é a Commedia Dell'Arte. A Commedia Dell'Arte era um espetáculo renascentista, mais literário do que teatral, que provocou uma reação dos atores italianos, levando-os a abandonar o drama - onde a narração substituíra a ação - e a criar um novo tipo de representação, com predomínio dos gestos expressivos e o mínimo de diálogo. O texto dramático literário foi substituído pelo *canevas*, tipo de plano, esboço ou roteiro no qual se destacavam somente os fatos principais, numa seqüência que possibilitava as livres improvisações dos comediantes [REV 87]. As personagens da Commedia eram fixas, o que mudava eram as situações. O corpo, a voz e a sensibilidade do comediante ajustavam-se de tal forma à personagem que o espectador esquecia-o, aplaudindo a personagem.

A improvisação está relacionada ao intuitivo e conseqüentemente com a espontaneidade [SPO 63] [SPO 99]. O intuitivo somente pode responder ao momento imediato, onde existe a liberdade de agir em um mundo que está em constante movimento. Este aspecto confere à improvisação o seu caráter de momentaneidade, no qual se encontrará todo e qualquer ato teatral. Isto indica que os fundamentos da improvisação teatral encontram-se na capacidade espontânea, simbólica e criativa do homem.

Deste modo, a idéia que geralmente se faz a respeito da improvisação é de algo informal, espontâneo, imprevisto, sem preparo prévio, inventado de repente, aleatório, enfim trata-se de um produto inspirado na própria ocasião e feito sem preparo. Em oposição tem-se a idéia do “ser em forma” como algo preparado, organizado, elaborado, deliberado, portanto um produto formalizado. No primeiro caso costuma-se inserir a improvisação e no segundo a representação. Contudo, estas definições não implicam que improvisação e representação sejam duas coisas distintas, elas são, na realidade, dois pólos de uma mesma matéria. A distância entre estes dois pólos, o improvisado e o representado, é que determina as diferenças entre si através de graus, onde a apresentação teatral se torna mais ou menos formalizada ou improvisada. Desta maneira, pode-se imaginar uma escala onde em um extremo está o teatro convencional, com ênfase na representação e permitindo o mínimo de improvisação, e no outro a improvisação, contendo o mínimo de representação.

Estes graus de improvisação se refletem nos tipos de aplicação da improvisação dentro dos componentes do espetáculo teatral. O espetáculo teatral é composto de três componentes principais, que estão presentes e compartilham um mesmo espaço: ator, texto e público. Cada um destes elementos pode determinar uma configuração improvisacional

na obra acabada, seja através da forma teatral ou através da apresentação de uma peça como um produto final.

A forma teatral é resultado de um processo voluntário de criação onde a espontaneidade e o intuitivo também exercem um papel de importância. Esta espontaneidade pode-se ser chamada de improvisação e é vista como algo inesperado que vai surgindo no decorrer da criação artística, ou seja, vai se manifestando durante os ensaios. Esta espontaneidade vai tomando forma e começando a fazer parte do processo voluntário. Desta maneira, a construção teatral vai escondendo as manifestações súbitas para dar lugar às ações elaboradas e formalizadas, o improviso deixa aparentemente de existir pois foi incluído na formalização.

A forma final do teatro somente se completa através da apresentação da peça para um público. Como esta apresentação é realizada “ao vivo”, em um determinado momento, nunca é possível se repetir exatamente a mesma coisa, pois um momento nunca é igual ao outro. Deste modo, a própria natureza momentânea do teatro já inclui um caráter improvisacional na obra acabada, por mais que esteja tudo preparado e ensaiado.

Estando o modelo teatral (ator e texto) formalizado, no momento em que será feita uma apresentação (exposição do modelo para uma platéia), cabe ao ator acionar os elementos do jogo cênico, apresentando-o com atualidade tanto para si quanto para quem o vê. Este simples fato faz com que o ator por mais que queira repetir a atuação programada e ensaiada, às vezes não o consiga. Isto não significa que o ator está improvisando e de fato não está, pois ele está representando um papel que foi altamente elaborado. Neste caso, a improvisação é dita como sendo um elemento implícito do ato teatral, pois a cada novo espetáculo o ator inclui algo novo. Isto ocorre na medida em que, no teatro formalizado, o ator se reveste de uma máscara, isto é a personagem que é pré-determinada pelo texto. O ator por sua vez interpreta a personagem sob o seu ponto de vista podendo também ser orientado por um diretor. O que interessa salientar é que a atuação implica o encontro de dois elementos, o ator (pessoa) e a personagem (ficção). O ator passa a jogar em cena com estes dois princípios e é neste momento que ocorre a improvisação.

Segundo Chacra [CHA 83], a configuração improvisacional dentro da reprodução do modelo do teatro formal está ligada ao fenômeno artístico que justifica a improvisação como inerente a todo o jogo cênico. Dando a este um caráter sempre novo na medida em que, pela sua própria natureza, não pode ser repetido perfeitamente igual todas as vezes que é representado. Quando algo de inesperado acontece no decorrer do espetáculo, por alguns segundos as ações tornam-se mais instintivas, mais improvisadas e desvinculadas do modelo formal pré-concebido. Nestes casos, a improvisação teatral torna-se um recurso com o qual o ator pode contar para superar obstáculos que possam interromper ou prejudicar o andamento da peça.

4.1 Elementos do Teatro Improvisacional

Assim como no teatro convencional, no teatro improvisacional existe a figura de um escritor ou autor da peça, de um produtor, de um diretor, de atores, do público e de outras pessoas responsáveis por dar suporte à criação, produção e apresentação de um espetáculo teatral. Nesta proposta iremos focar quatro destes elementos: o texto, produzido por um escritor ou autor, o ator, o público e o diretor. Estes quatro elementos foram escolhidos pois

estão diretamente presentes no momento da apresentação teatral, no momento em que ocorre a improvisação. Nesta seção iremos abordar os três primeiros elementos e deixaremos as discussões sobre o diretor para o próximo capítulo.

4.1.1 O Texto

O texto teatral pode ser entendido como um conjunto de sinais, signos e símbolos – verbais e não verbais - existentes durante um espetáculo, ou como uma peça escrita para teatro. Na passagem do texto escrito no papel para a cena em um espetáculo, as palavras que constituem os diálogos das personagens sofrerão alterações de significado, na medida em que os atores tenderão a misturá-los com outros símbolos – não verbais – formando o texto teatral propriamente dito. Alguns atores memorizam suas falas, gestos e entonações vocais repetindo quase da mesma maneira sua interpretação a cada apresentação. Deste modo, encontram menos possibilidade de improvisarem, ao contrário de seus colegas que deixam se levar pelo momento, fazendo da improvisação o seu modo de expressão. Estes atores partem do princípio de que a peça pré-estabelecida é um pretexto para a atuação e por este motivo são chamados de autores atores.

4.1.2 O Ator

Segundo Chacra [CHA 83], o foco de uma peça improvisada está no ator. O ator está sempre improvisando, não há como escapar de uma arte cuja essência é a qualidade momentânea. Um ator nunca se repete mesmo que ele deseje, pois é impossível uma reprodução idêntica da atuação dada a própria natureza do teatro.

O caráter fundamental da improvisação é a espontaneidade e esta é a base da arte do ator: a arte da flexibilidade, do improviso e das surpresas, mas também do controle e da adaptação. Ao mesmo tempo em que o ator possui liberdade para improvisar, suas improvisações devem ser organizadas, coerentes e devem refletir um estado de espontaneidade interior relacionado com a sua função dentro da peça. Cada papel oferece ao ator uma oportunidade de improvisação, colaboração e co-criação entre ator e diretor. Isto não significa que o ator vá criar novas falas ou modificar os objetivos propostos pelo diretor. Pelo contrário, as falas pré-definidas para um papel e as instruções e orientações do diretor constituem a base sobre a qual o ator desenvolve suas improvisações. Desde modo, o ator pode improvisar como ele declama suas falas e como executa suas instruções. Isto ocorre devido à interpretação que o ator faz da personagem, englobando suas características físicas (externas) e psicológicas (internas) [CHE 86].

Chekhov [CHE 86] salienta que para o ator realizar uma atuação improvisada, ele deve tentar definir somente seus estados de espírito e não o tema ou enredo envolvido na atuação. O ator deve atuar de acordo com seus sentimentos, emoções e estados de espírito do momento. Desta maneira, cada momento sucessivo de uma improvisação será um resultado psicológico (e não lógico) do momento que o precedeu.

A improvisação está presente em toda a interpretação através de graus que definem uma configuração mais ou menos formal e improvisada à interpretação. Estes graus de improvisação proporcionam a existência de dois tipos de improvisação: improvisação limitada à via traçada e improvisação livre. No primeiro caso o ator é visto como um intérprete do texto e de uma personagem, e a improvisação ocorre implicitamente durante o espetáculo. No segundo caso, o ator é visto como um intérprete-criador, ele substitui o

dramaturgo e é responsável por criar e executar tanto o texto quanto o personagem no decorrer da atuação. Os intérpretes-criadores são os mais sensíveis à reação do público. Seus improvisos são fruto de suas habilidades específicas e do convívio direto com o espectador durante o ato de representação.

4.1.2.1 Improvisação em Grupo ou Improvisação Coletiva

Segundo Chekhov [CHE 86], a improvisação em grupo/coletiva é diferente da improvisação individual. Na improvisação em grupo/coletiva, além de cada ator definir seu estado de espírito, devem ser definidas pelo menos as linhas gerais de um tema e deve ser realizada a distribuição dos papéis que serão encenados.

De acordo com Chekhov [CHE 86], o exercício de improvisação em grupo/coletiva ocorre da seguinte maneira: inicialmente o grupo deve escolher uma cena de uma peça que nenhum dos membros tenha atuado e que nenhum deles tenha visto no palco ou tela; logo após os papéis devem ser distribuídos e deve-se escolher um dos membros para “diretor” e solicitar que ele encene o começo e final da cena escolhida; depois, os atores, conhecendo o conteúdo da cena, devem improvisar toda a parte intermediária. Tendo chegado ao final da cena, os atores devem solicitar ao diretor que o mesmo encene mais uma vez uma pequena seção da cena, mais ou menos no meio e por último os atores devem começar a improvisação de novo, desde o início até o ponto dirigido pelo diretor e encenar de maneira improvisada até o final.

4.1.3 O Público

Historicamente falando, durante as primeiras apresentações teatrais a platéia participava ativamente do ato teatral. Depois passou a exercer um papel passivo, simplesmente assistia a atuação dos atores. Atualmente está se retomando a idéia de que a platéia deve participar do ato teatral. É desta maneira que surge a improvisação coletiva (atores e platéia interagindo para a construção da peça). No improviso coletivo, o talento improvisacional não se encontra somente no ator, mas é dividido entre artistas e não-artistas.

4.2 Agentes Improvisacionais

O conceito de agentes improvisacionais surgiu com o paradigma da improvisação dirigida [HAY 94a] proposta pelo projeto Teatro Virtual (seção 3.1). Este paradigma se baseou em idéias de improvisação providas do Teatro Improvisacional, como as apresentadas na seção anterior.

4.2.1 Improvisação Dirigida

A improvisação dirigida pressupõe a existência de um mundo virtual habitado por atores animados. Estes atores animados são na realidade agentes cuja função é interagir com o usuário, a fim de que juntos realizem alguma tarefa. Hayes-Roth [HAY 98b] define atores animados como criaturas de computador com mentes de software, que se assemelham aos atores reais e fictícios numa série de dimensões qualitativas e funcionais. Por exemplo, eles possuem funções cognitivas básicas para percepção e ação dentro do seu contexto de aplicação, expressam seus comportamentos através de diferentes maneiras

(como movimentos, gestos e conversas) e diferentes mídias (como gráficos, animações, voz e texto), podem possuir qualidades psicológicas como personalidade, emoção, motivação e relacionamentos sociais, podem possuir diferentes tipos de conhecimento e utilizá-los de diferentes maneiras dentro da aplicação e podem encenar diferentes papéis em diferentes tipos de aplicações.

Na improvisação dirigida, os atores trabalham dentro de restrições de direções que variam o grau com o qual elas restringem o comportamento dos atores. As direções podem ser abstratas e deixar o ator quase que completamente livre ou podem ser bastante específicas, descrevendo os menores detalhes de comportamento. Elas podem restringir qualquer aspecto do comportamento incluindo o papel, caracterização e ações do ator. As direções podem ser informadas na forma de roteiros completos, ou podem ser informadas interativamente, durante a execução. Estas direções podem vir de diversas fontes, incluindo pessoas, outros componentes de computador ou os próprios atores. Como Laurel [LAU 93] salienta, quando se pede a uma pessoa que seja criativa sem nenhuma direção ou restrição, o resultado é um sentimento de falta de poder ou até mesmo a completa paralisia da imaginação. A imposição de restrições enfoca o processo criativo, aumentando o poder imaginativo através da redução das opções possíveis.

Embora sejam possíveis vários modos de atuação improvisacional, como visto nas seções anteriores, na improvisação dirigida foi focalizado o modo clássico praticado pela *Commedia dell'Arte* da renascença italiana e modernamente adaptada por grupos de teatro em Chicago. Nesta abordagem os personagens são associados a papéis padrões em um cenário familiar, mas improvisam os detalhes das suas atuações.

Deste modo, os atores são definidos por três construtores dramáticos: enredo, papel e personagem [HAY 97a]. Enredo é uma seqüência de ações temporariamente restringida envolvendo um conjunto de indivíduos. O enredo e suas ações constituintes podem ser um pouco abstratos. Por exemplo, o protótipo de um enredo é: a conhece b, a se apaixona por b, a perde b, a reconquista b. Papel é a classe de indivíduos cujos comportamentos, relações e interações são conhecidas tanto dos atores quanto da audiência. Por exemplo, um enredo é instanciado com os seguintes papéis: o garoto apaixonado e a garota que ele ama. O personagem representa uma personalidade definida como uma configuração coerente de traços psicológicos. Por exemplo, um personagem dentro de um cenário pode ser: tímido e sensível, sociável e rude ou bobo e fingido. São os personagens, com suas personalidades distintas, que provocam reações emocionais nas pessoas e que as fazem amá-los ou odiá-los. O enredo, papel e o personagem são substancialmente independentes e podem ser dirigidos ou improvisados separadamente. Isto significa que os atores humanos podem ser dirigidos em qualquer um destes conjuntos e deixados livres para improvisação em outro. Apesar dos construtores serem substancialmente independentes, consideramos que um ator (agente) pode interpretar um ou mais personagens, e cada personagem pode estar relacionado a um ou mais papéis que serão assumidos no momento da execução. Desta maneira, os atores estão indiretamente relacionados aos papéis que os seus personagens executam.

A improvisação pode ocorrer em qualquer um dos três construtores dramáticos (enredo, papel e personagem) e pode ser restringida através de direções. Deste modo, cada um dos construtores pode receber restrições de direção que irão influenciar na sua improvisação.

4.2.2 Características dos Agentes Improvisacionais

Hayes-Roth [HAY 95b] apresenta os seguintes requisitos para um agente improvisacional:

- deve ter capacidades funcionais de percepção, raciocínio e ação em ambientes em tempo-real complexos e incertos. Estas capacidades são modeladas em um certo grau na maioria dos agentes convencionais. Contudo, o agente improvisacional precisa integrar uma grande variedade destas capacidades. Os agentes devem continuamente monitorar e interpretar o(s) comportamento(s) de outro(s) agente(s), monitorar o seu ambiente e controlar os seus próprios comportamentos, a fim de seguir o plano corrente, enquanto mentalmente improvisam as direções internas que elaboram as direções dos usuários e devem, também, adaptar o comportamento dos seus parceiros de modo que o plano seja seguido corretamente;
- deve seguir as instruções abstratas (por exemplo, ser curioso, amigável) que requerem interpretação e iniciativa a fim de determinar o curso de uma ação;
- deve exibir inteligência cotidiana, deve possuir um senso comum sobre o mundo físico, um modelo psicológico simples e convenções sociais;
- deve exibir comportamento vivo, ou seja, pode variar a interpretação de uma determinada instrução dependendo da ocasião, pode executar comportamentos específicos em diferentes ocasiões e pode ter emoções associadas com seu comportamento e interações;
- deve colaborar de perto com outros personagens e compartilhar controle. A maioria dos paradigmas de sistemas multiagentes possuem interesse em que os seus agentes trabalhem juntos e limitem as interações durante a execução de atividades coletivas. Na verdade, uma boa parte do trabalho em sistemas multiagentes volta sua atenção para as questões de negociação e planejamento de interações. Contudo, os bons improvisadores não negociam ou planejam as suas interações, eles simplesmente cooperam da melhor maneira possível;
- deve exibir perícia improvisacional. A maioria dos trabalhos em agentes inteligentes enfoca a competência do agente para executar algum processo e, especialmente, o seu sucesso em alcançar os objetivos. A improvisação dirigida, ao contrário, é explicitamente orientada a processos, ou seja, a instrução de comportamento fornecida ao agente-ator é o seu objetivo. Não se pode dizer que a sua execução está “correta” ou “incorreta”, pois ele pode executar de diferentes maneiras as restrições das instruções dos usuários, sendo que todas levam a uma execução “correta”.

Os agentes improvisacionais fazem uso de muitas habilidades intelectuais tradicionais estudadas como componentes da Inteligência Artificial. Contudo, acredita-se que os componentes utilizados necessitam de alguma revisão para que sejam aplicados a agentes deste tipo. Segundo Hayes-Roth e Doyle [HAY 98b] os componentes tradicionais da IA que devem ser modificados e aperfeiçoados são: planejamento, aprendizagem e processamento da linguagem natural. Para Hayes-Roth e Doyle [HAY 98b] planejar se torna improvisar, aprender se torna relembrar e processamento da linguagem natural se torna conversação.

4.3 Considerações sobre o Capítulo

Improvisação é algo espontâneo relacionado às reações momentâneas a uma determinada situação. Isto significa que a improvisação pode estar presente em todos os momentos da nossa vida. Esta presença é caracterizada através de graus que podem ir desde a completa improvisação de nossas ações até o mínimo de improvisação que é realizada de maneira inconsciente pela própria natureza humana.

Como salientado por Laurel [LAU 93], uma liberdade completa de ação por parte das pessoas pode dificultar o processo criativo e de tomada de decisão. As pessoas precisam ser restringidas para que possam direcionar seu processo de criação. Assim como as pessoas precisam ser restringidas, entendemos que os agentes improvisacionais também precisam ser restringidos, pois se forem deixados à mercê da livre improvisação, podem não chegar a lugar nenhum. Hayes-Roth apresenta uma primeira abordagem para esta restrição através do paradigma da improvisação dirigida, onde podem ser realizadas restrições nos três construtores dramáticos básicos: enredo, papel e personagem. Esta abordagem é bastante interessante, pois apresenta uma possibilidade de direção de atores animados que é condizente com a fundamentação do teatro improvisacional. Contudo, a restrição de direção fornecida aos atores fica a cargo de diretores humanos, que podem ou não ter conhecimento e considerar as questões relativas ao diretor improvisacional.

Toda a ênfase do trabalho de Hayes-Roth está nos processos realizados pelos atores, não mencionando os processos realizados pelos diretores. É mencionado que agentes humanos realizam a direção de atores, mas não é tratado nenhum aspecto de como esta direção é realizada, se é realizada também de uma maneira improvisada, considerando o papel do diretor no teatro improvisacional. Um dos pontos do nosso trabalho se concentra justamente neste aspecto. Queremos incorporar as funcionalidades de um diretor improvisacional humano para um diretor improvisacional de software que tenha a capacidade de coordenar e dirigir um grupo de atores de software, realizando, assim, uma coordenação improvisacional.

Para trabalharmos esta questão precisamos também estudar um outro ponto importante do nosso trabalho, que é uma questão mais profunda relacionada à própria improvisação e à maneira como ela pode ser abordada na computação em geral e mais especificamente na Inteligência Artificial. A improvisação foi inserida na Inteligência Artificial através dos conceitos de improvisação dirigida e agentes improvisacionais propostos por Barbara Hayes-Roth. Contudo, Hayes-Roth [HAY 98b] propõe que agentes improvisacionais são uma nova categoria de agentes e esta categorização está mais baseada em aspectos artísticos do que em aspectos computacionais. Acreditamos que agentes improvisacionais não são uma nova categoria de agentes, mas são um tipo de agente deliberativo para resolução de problemas relacionados à improvisação. Neste sentido, iremos formalizar a idéia de agentes improvisacionais como um tipo de agente deliberativo e mostrar claramente como suas características artísticas podem ser mapeadas computacionalmente. Para tanto, iremos estudar uma arquitetura específica de agentes deliberativos, a arquitetura BDI [WOO 2000], e mostraremos como a improvisação pode ser considerada dentro de uma arquitetura deste tipo.

Como os aspectos de improvisação a serem abordados nos agentes improvisacionais como agentes deliberativos, englobam a direção improvisacional, no próximo capítulo apresentaremos a fundamentação teórica relacionada ao diretor no teatro improvisacional.

5 PAPEL DO DIRETOR NO TEATRO IMPROVISACIONAL

O diretor é uma figura importante no meio artístico, seja este teatro ou cinema, pois é responsável por coordenar vários processos envolvidos na produção de uma peça de teatro ou filme. Ele é responsável por selecionar as pessoas envolvidas na produção do cenário e das vestimentas, é responsável por selecionar os atores, interagir com o escritor ou autor a fim de obter a essência da apresentação e repassá-la aos atores. Todos os diretores de alguma maneira realizam estes e outros processos. Neste trabalho iremos focar um tipo específico de direção que é a realizada pelo diretor criativo ou diretor improvisacional.

5.1 Características de um Diretor Criativo ou Improvisacional

Morrison [MOR 74] salienta que o diretor possui os talentos de um artista criativo e não é simplesmente um organizador de atores e cenas, mas alguém que coloca a peça em um processo imaginativo. O diretor leva a peça à vida em um estágio que o texto escrito não pode: ele aperfeiçoa as virtudes, distingue as falhas e limitações e ajuda um grupo de atores a representar isto com sentimento e emoção.

De acordo com Black [BLA 91] algumas das características que um diretor deve possuir para trabalhar em um modo criativo são:

- habilidades para análise textual, performance dos ritmos e performance da estrutura da peça;
- imaginação, ou seja, a capacidade para encontrar e comunicar a imagem (por analogia ou metáfora) e invocar estas imagens no teatro;
- sentido colaborativo resumido como uma habilidade essencialmente interpessoal, incluindo a habilidade para usar os recursos e técnicas de atores, projetistas e técnicos;
- habilidade para dirigir as atuações criativas dos atores;
- habilidades organizacionais e executivas, incluindo a habilidade de tratar com problemas de relações administrativas e públicas bem como problemas artísticos e
- ego, temperamento artístico que lhe dá a segurança necessária para tomar decisões e em alguns casos defender julgamentos criativos.

Cohen e Harrop [COE 74] e Black [BLA 91] salientam que o diretor geralmente inicia seu trabalho escolhendo a peça a ser apresentada, isto deve ser definido considerando limite de tempo e espaço. O diretor deve ter em mente que cada elemento do evento teatral tem significância e neste sentido deve se selecionar os atores, roupas e cenário adequado para a realização da peça. Além disso, o evento teatral é único e existe somente no presente, ou seja, uma determinada atuação nunca pode ser replicada. Sempre irá estar presente o momento de espontaneidade, ou seja, a improvisação.

Morrison [MOR 74], Black [BLA 91] e Berger [BER 98] salientam que uma das funções mais importantes do diretor é interpretação de uma peça e o treinamento e coordenação de atores. Este processo de treinamento e coordenação pode possibilitar ao diretor a sugestão de uma linguagem corporal ou comportamento para os atores a fim de adicionar força à cena tornando-a clara, unificada e coerente, mas deixando uma liberdade para os atores experimentarem a sua própria imaginação e sensibilidade.

Spolin [SPO 63] [SPO 99] considera que a primeira tarefa que um diretor improvisacional deve realizar para coordenar seus atores é informá-los do que deve ser feito durante a encenação da peça. Esta informação é passada para os atores através de *scripts*. Estes *scripts* somente devem conter as direções que levam a algum tipo de ação ou diálogo. É importante que o diretor dê liberdade para que seus atores possam atuar espontaneamente. Para construir os *scripts* o diretor deve interagir com o escritor da peça, obtendo informações sobre a peça, interpretando-as e inserindo sua visão criativa particular.

Além de fornecer informação inicial para os atores realizarem uma atuação, o diretor deve auxiliar os atores a encontrar soluções para problemas que possam surgir durante uma atuação e deve também, ao final de uma atuação, realizar uma avaliação tanto da performance dos seus atores quanto dele próprio como diretor [SPO 99].

Levando em conta todas as características apresentadas anteriormente, observamos que as principais funções de um diretor criativo ou improvisacional são: construção de *scripts*, avaliação e auxílio na solução de problemas. O estudo destas funções é importante, pois nos fornece uma base teórica necessária para o entendimento dos processos do diretor, o que é de fundamental importância para o desenvolvimento de um processo automatizado de direção.

5.2 Construção do Script

Segundo Black [BLA 91], o diretor é um artista que cria uma obra de arte. Para realizar a obra de arte chamada peça teatral, o diretor, partindo do *script* original do autor, utiliza improvisação para criar um novo impulso ou uma nova direção para o *script*. Este novo impulso é a interpretação que o diretor faz do *script*.

Cohen [COH 74] salienta que é comum assumir que a interpretação do *script* é a primeira responsabilidade do diretor, porque todas as outras decisões de direção, podem ser respondidas somente em termos de uma interpretação pré-formulada. A interpretação do diretor é um conjunto de idéias, imagens e sentimentos que expressam o que o diretor quer que a sua peça comunique para a audiência. Este conjunto de idéias inclui as concepções do diretor do significado da peça e qualquer significado adicional que ele queira incluir na produção.

Os diretores geralmente iniciam a interpretação de uma produção com uma idéia clara e objetiva da idéia central e procuram tornar cada momento da peça em conformidade com aquela interpretação geral. Os diretores fazem uso de informações contidas no texto e no seu mundo para construir a sua interpretação do *script*.

5.3 Avaliação

Segundo Black [BLA 91], a habilidade de perceber e avaliar o trabalho realizado por um diretor é indispensável para o progresso do diretor artístico. Neste sentido, o diretor deve ser capaz de realizar uma avaliação tanto dos seus atores quanto da sua própria atuação. Uma das coisas mais importantes para a tarefa da avaliação é o poder da observação das ações dos atores e das suas próprias ações.

Spolin [SPO 63] [SPO 99] salienta a importância da avaliação e da observação, quando ele menciona que o diretor sempre ao final de um exercício improvisacional ou de uma performance deve realizar uma avaliação dos seus atores. Spolin [SPO 99] propõe que o diretor deve avaliar o problema de atuação objetivamente. Deste modo, ele tenta responder as seguintes questões: O problema foi resolvido? Os atores agiram ou reagiram? Eles deixaram alguma coisa acontecer?, Eles permaneceram com o problema? Estas questões podem ser interpretadas da seguinte maneira: a primeira questão considera se os atores atuaram o *script* que lhes foi passado, ou seja, se eles atingiram seus objetivos. A segunda e terceira questões estão relacionadas à capacidade de improvisação dos atores e verifica se eles foram capazes de improvisar comportamento, agindo sobre o mundo e não simplesmente reagindo a ele, e se eles estavam abertos a novas oportunidades. A última questão verifica se os atores executaram suas atividades iniciais, não perdendo o rumo e atingindo outros objetivos que não foram especificados pelo diretor através da tradução do *script* fornecido pelo autor. O sucesso, ou não, do diretor está diretamente relacionado à atuação dos atores, pois se os atores atingiram os objetivos esperados a direção fornecida deve ter sido eficiente.

5.4 Resolução de Problemas

Segundo Berger [BER 98], às vezes o diretor pode ter que ajudar os atores a encontrar a direção dos seus personagens; e, mais frequentemente, o diretor pode ter que ajudar os atores a projetar o impulso que motiva a audiência.

Segundo Spolin [SPO 63] [SPO 99], o diretor pode atuar como um professor que deve guiar a preparação do grupo de atores quando necessário. O diretor-professor pode achar uma ou mais maneiras de resolver um problema imediato. Problemas imediatos podem surgir e são identificados pelos atores e comunicados ao diretor para que o mesmo providencie uma solução para o problema.

5.5 Considerações sobre o Capítulo

Vários autores salientam a importância do diretor como um coordenador dos seus atores e dos processos envolvidos na criação de uma peça teatral. Os principais aspectos

envolvidos na direção, e conseqüentemente na coordenação, são a construção do *script*, a avaliação do trabalho realizado pelos atores, e do seu próprio trabalho como diretor, e a resolução de problemas quando os mesmos aparecem. Todos estes aspectos devem ser considerados na construção do nosso processo de direção improvisacional. Outro fator a ser considerado é que este processo de direção improvisacional pode tanto ser realizado por um agente específico, o diretor improvisacional, ou por um agente representando um ator. O último caso enquadra-se no que Chekhov [CHE 86] chama de improvisação coletiva ou de grupo (seção 4.1.2.1), onde cada ator é responsável por ser seu próprio diretor e um dos atores é escolhido como “diretor geral” para informar as linhas gerais de atuação e avaliar a performance ao seu final. Estes dois tipos de direção propostas pelo teatro improvisacional podem ser entendidas como tipos de coordenação improvisacional centralizada e descentralizada de atores e serão considerados na arquitetura de agentes improvisacionais como agentes deliberativos a ser apresentada nesta tese.

A partir dos conceitos de improvisação apresentados nos capítulos 4 e 5, podemos identificar diferentes tipos de improvisação, que serão apresentados no próximo capítulo.

6 TIPOS DE IMPROVISACÃO

Embora a idéia que geralmente se possui sobre improvisação é de algo informal, espontâneo, aleatório, inspirado na própria ocasião e sem preparo, este é um dos extremos da improvisação. Na realidade, a improvisação possui graus que vão desde algo completamente aleatório a algo onde a improvisação foi incorporada na representação formal através dos ensaios. Desta maneira, poderíamos considerar que um extremo da improvisação seria a execução de algo completamente aleatório, e outro extremo seria algo completamente planejado, onde a improvisação foi incorporada na estrutura previamente planejada das possíveis ações futuras. Um nível intermediário seria a existência de um curso de ação abstrata na forma de intenções, onde as ações seriam escolhidas considerando a situação que o agente se encontra no momento da execução. Dependendo do nível de abstração, o agente estará mais voltado para a improvisação ou para o planejamento.

Os graus de improvisação se refletem nos tipos de aplicação da improvisação dentro dos componentes principais do espetáculo teatral: ator, texto e público. A improvisação no ator está relacionada à maneira como ele interpreta a personagem, considerando as suas características físicas e psicológicas. A improvisação no texto ocorre através do conjunto de sinais, signos e símbolos, verbais e não verbais que são deixados em aberto para a interpretação do ator. Na improvisação do público a platéia interage com os atores para a construção da peça.

Desta maneira, tendo como base a fundamentação teórica do teatro improvisacional, apresentada nos capítulos 4 e 5, podemos constatar a existência de quatro tipos de improvisação:

1. improvisação de texto;
2. improvisação de ator ou improvisação de personalidade;
3. improvisação do público;
4. improvisação de solução de problemas.

Os tipos de improvisação podem ocorrer em dois momentos distintos: durante a preparação ou construção dos elementos envolvidos em uma apresentação e durante a execução da apresentação. O momento da preparação envolve um processo de deliberação

do agente. Dependendo do momento em que ocorre, um tipo de improvisação pode estar relacionado a diferentes agentes, a saber, diretor ou ator.

Tabela 6.1: Tipos de Improvisação

Tipo de Improvisação	Momento	Agente(s) envolvido(s)
Texto	Preparação	Diretor
	Execução	Ator
Personalidade	Preparação	Diretor e Ator
	Execução	Ator
Público	Execução	Ator
Solução de Problemas	Execução	Diretor e/ou Ator

A tabela acima merece algumas considerações. No caso da improvisação do texto no momento da preparação está envolvido o diretor. O diretor pode ser um agente diferente do ator no caso de termos uma construção de apresentação centralizada, mas o diretor pode ser o próprio ator quando a apresentação é construída de forma conjunta com outros atores em um processo descentralizado. Quando a improvisação ocorre no momento da execução do texto, quem está envolvido é o ator, pois ele irá utilizar as descrições da sua personalidade para execução da ação mais apropriada. A presença do diretor na improvisação da preparação da personalidade e na execução da solução de problemas ocorre quando existir um processo centralizado. Caso contrário, o próprio ator assume o papel de diretor. A improvisação de público e de solução de problemas não possui uma fase de preparação, pois os agentes não conhecem antecipadamente as reações do público e os problemas que podem surgir durante a execução. Entendemos que os problemas e reações conhecidas pelo agente já foram incorporados no seu texto no momento da preparação do texto. Contudo, entendemos que a improvisação de público e de solução de problemas possui uma estrutura base que possibilita a improvisação no momento em que algo é detectado durante a execução.

A palavra problema no tipo de improvisação solução de problemas é entendida como algo gerado pela falta do enquadramento de uma situação em alguma descrição de comportamento abstrato do agente, ou seja, o agente não possui um curso de ação abstrato que se enquadre em uma determinada situação.

6.1 Categorização da Improvisação em Cada um dos Tipos de Improvisação

Nesta seção iremos apresentar o que se entende por improvisação em cada uma das fases presentes nos tipos de improvisação.

6.1.1 Improvisação de Texto

6.1.1.1 Fase de Preparação

Fornece um curso de ação abstrato, não definindo a ação até o nível da ação primitiva. Por ação primitiva se entende uma ação pronta para ser executada. O curso de ação abstrato fornece alternativas que serão escolhidas no momento adequado, ou seja, no momento da execução, ou no momento que imediatamente precede a execução. A idéia por trás do curso de ação abstrato é a de planos como intenções e não planos como programas. De acordo com Pflieger e Hayes-Roth [PFL 98] e Agre e Chapman [AGR 88a] na visão de planos como programas, um plano é um programa executável consistindo de ações primitivas que o agente executa a fim de agir. Nesta abordagem o planejamento é um tipo de programação automática e a realização do plano consiste na sua execução direta. Na outra visão, o plano é um comprometimento a um objetivo que guia, mas não determina as ações específicas que o agente deve executar. Desta maneira, o agente não pode executar diretamente seu plano, ele pode executar comportamentos que podem estar mais ou menos consistentes com seus planos.

6.1.1.2 Fase de Execução

A execução do texto está diretamente relacionada à personalidade do personagem que o ator representa no momento, ou seja, a execução da personalidade do ator. Nesta fase, o ator deve escolher dentro de um conjunto de ações possíveis a ação mais apropriada considerando a situação do momento, seu conhecimento, suas habilidades e sua personalidade. A improvisação está na função de escolha da ação a ser executada. Para a função envolver improvisação ela deve considerar as características da situação do momento.

6.1.2 Improvisação da Personalidade

6.1.2.1 Fase de Preparação

Esta fase envolve a descrição de diferentes personalidades para um personagem. As personalidades são relacionadas a possíveis situações que o agente pode se encontrar. Deve-se possuir especificação de várias instâncias de personalidade, ou seja, maneiras de expressar uma mesma personalidade. Estas instâncias devem estar relacionadas a ações verbais e não verbais (físicas).

6.1.2.2 Fase de Execução

Nesta fase, deve-se escolher, através de uma função de ativação, uma entre as várias ações para representar a personalidade. A ação escolhida representa a ação mais adequada para um determinado momento. Tendo escolhido a ação, deve-se escolher qual instância executar. A escolha da instância pode ser uma primeira vez aleatória e posteriormente ter uma função para não deixar escolher aleatoriamente sempre a mesma instância. A função deve tratar da questão da ilusão de vida e variabilidade de comportamento.

6.1.3 Improvisação do Público

6.1.3.1 Fase de Preparação

Não existe preparação da reação do público à apresentação, pois é algo que acontece durante a execução da apresentação.

6.1.3.2 Fase de Execução

As reações do público a uma apresentação formam a improvisação do público a uma determinada apresentação. Esta reação e/ou interação pode provocar improvisação nos atores. Por exemplo, se a platéia não está contente com a performance, os atores devem improvisar, fazendo uso de uma veia cômica para deixar a platéia mais contente.

6.1.4 Improvisação de Solução de Problemas

6.1.4.1 Fase de Preparação

Não existe preparação, pois não se sabe os problemas que podem surgir. Os possíveis problemas conhecidos podem ser abordados na improvisação do texto, descrito na forma de um curso de ação abstrato.

6.1.4.2 Fase de Execução

Esta fase deve abordar o que fazer quando se encontra frente a um problema que surge durante a execução e para o qual não existe curso de ação abstrato associado. Esta fase está relacionada a improvisação explícita [CHA 83] que indica como solucionar problema imediatos, ou seja, como solucionar imprevistos.

Para entender o tipo de problema que queremos solucionar nesta fase, é necessário apresentar a nossa visão sobre os tipos de situações e eventos que podem acontecer. Uma situação ou evento pode ser de quatro tipos: imprevisto, previsível, desconhecido ou conhecido. Um evento imprevisto pode ser de dois tipos: um acidente, ou seja, um evento que ocorre por acaso, devido a uma coincidência e algo desconhecido, ou seja, um evento que ocorre, mas cuja causa é desconhecida.

Iremos tratar a solução de problemas como uma situação ou um evento imprevisto do tipo acidente como o que ocorre na improvisação explícita do teatro improvisacional. Entendemos que um evento imprevisto desconhecido pode ser abordado através do uso de técnicas de descoberta e um evento imprevisto cuja causa é conhecida, pode ser abordado através do uso de técnicas de improvisação. A tabela 6.2 mostra os tipos de eventos e os mecanismos utilizados para solucioná-los.

Tabela 6.2: Tipos de Eventos e Mecanismos para Solucioná-los

	Conhecido	Desconhecido
Previsto	Planejamento e Improvisação Implícita	-----
Imprevisto	Improvisação Explícita	Descoberta

Através da tabela 6.2 podemos observar que uma situação ou evento conhecido e previsto pode ser solucionado através de técnicas de planejamento e improvisação implícita. Por planejamento entendemos as técnicas tradicionais de planejamento, ou seja, a visão de planos como programas. Por improvisação implícita entendemos a idéia do agente possuir um curso abstrato de comportamento, ou seja, a abordagem de planos como intenções. Um evento imprevisto, mas cuja causa é conhecida, pode ser solucionado através de técnicas de improvisação explícita. Não existe um evento que seja ao mesmo tempo conhecido e imprevisto. Um evento imprevisto, mas cuja causa é desconhecida, pode ser solucionado através de técnicas de descoberta.

O tratamento da solução de problemas como improvisação explícita é uma novidade no uso da improvisação na computação. A questão da solução de problemas como improvisação explícita será apresentada em maior profundidade no capítulo 10, onde iremos apresentar uma possível abordagem para esta questão envolvendo analogia por similaridade.

A fim de mostrar a existência de algum tipo de improvisação em arquiteturas BDI e, desta maneira, fazer o relacionamento de agentes improvisacionais com agentes deliberativos que usam arquitetura BDI, no capítulo 8 serão apresentadas algumas arquiteturas BDI. No capítulo 9 analisaremos estas arquiteturas a fim de identificar os tipos de improvisação presentes nas mesmas.

7 DIREÇÃO COMO COORDENAÇÃO EM SISTEMAS IMPROVISACIONAIS

Coordenação é uma das várias características importantes que agentes vivendo em uma sociedade devem ter, a fim de colaborar com outros agentes. A coordenação está em todos os lugares. Por exemplo, um professor usa coordenação na sua sala de aula para gerenciar seus alunos, um gerente usa coordenação para organizar quais atividades seus subordinados irão executar, um chefe de cozinha usa coordenação para indicar quais pratos seus assistentes devem preparar e assim por diante. Podemos ir além destes exemplos e abstrair que cada um executando um processo de coordenação pode ser visto como um diretor de teatro que tem que organizar e dirigir uma peça. Assim como no teatro, as pessoas se comportam de acordo com a situação momentânea e podem expressar comportamentos interessantes e surpreendentes. Desta maneira, propomos o uso das idéias do Teatro Improvisacional para suportar um processo particular de coordenação de agentes, chamado de direção improvisacional. Como a própria fundamentação teórica sugere, esta coordenação será baseada em improvisação ao invés das técnicas tradicionais de planejamento.

A fim de entendermos como a direção é vista como um processo de coordenação improvisada e não planejada, iremos, neste capítulo, apresentar questões relativas à coordenação e planejamento e iremos mostrar quais mudanças são necessárias para a construção da direção improvisacional em um sistema.

7.1 Coordenação em Sistemas Multiagentes

Coordenação é uma questão central nos sistemas de agentes de software em particular e na inteligência artificial distribuída em geral. Segundo Nwana et al [NWA 96], coordenação é um processo no qual agentes se engajam a fim de garantir que uma comunidade de agentes individuais aja de maneira coerente. Coerência se refere a quão bem o sistema de agente se comporta como uma unidade.

Existem várias razões pelas quais os agentes precisam ser coordenados:

- prevenir a anarquia ou caos: coordenação é necessária ou desejável porque, com a descentralização nos sistemas baseados em agentes, a anarquia pode ocorrer facilmente. Para realizar tarefas comuns um grupo de agentes precisa estar coordenado;
- atingir restrições globais: geralmente existem restrições globais que um grupo de agentes deve satisfazer se eles querem ter sucesso. Assim, os agentes precisam coordenar seus comportamentos se eles querem atingir restrições globais;

- especialidade, recursos e informação distribuída: como existe distribuição de especialidade, recursos e informação, os agentes devem se coordenar para trabalhar em conjunto;
- os objetivos dos agentes são freqüentemente interdependentes, existindo dependência entre as ações dos agentes, as atividades dos agentes devem ser coordenadas;
- eficiência: embora um agente possa realizar uma função de maneira independente, a coordenação é necessária pois a informação descoberta por um agente pode ser usada por outro. Isto faz com que a realização das tarefas de ambos agentes seja mais rápida.

O resultado do processo de coordenação é um conjunto de agentes coerentes, ou seja, um conjunto onde *deadlock* e *livelock* [HUH 99] são evitados. *Deadlock* se refere ao estado no qual ações futuras entre dois ou mais agentes são impossíveis; *livelock* se refere aos cenários onde agentes continuamente executam, mas nenhum progresso é realizado. Um conjunto coordenado também maximiza as capacidades dos agentes individuais e minimiza conflitos e contenção de recursos entre eles. Um sistema coerente irá minimizar ou evitar esforços conflitantes ou redundantes entre os agentes. Todos estes aspectos deixam claro que a coordenação é uma propriedade necessária e desejável em sistemas de agentes.

Segundo Huhns e Stephens [HUH 99], coordenação é uma propriedade de um sistema de agentes que executa alguma atividade em um ambiente compartilhado. O grau de coordenação é a extensão para a qual os agentes evitam atividades extras pela rejeição de *deadlock* e *livelock* e manutenção das condições de segurança do sistema. A coordenação pode ser dividida em dois tipos: cooperação e competição. Ao definir coordenação, Huhns e Stephens [HUH 99] falam em cooperação e negociação, eles não definem competição. Iremos fazer uma reflexão sobre isto na seção 7.3. Desta maneira, para os autores cooperação é coordenação entre agentes não antagônicos, enquanto a negociação é coordenação entre agentes competitivos ou agentes egoístas (figura 7.1). Tipicamente para que a cooperação tenha sucesso é necessário que os agentes possuam modelos dos outros agentes, o que implica na sociabilidade dos agentes que fazem parte da sociedade.

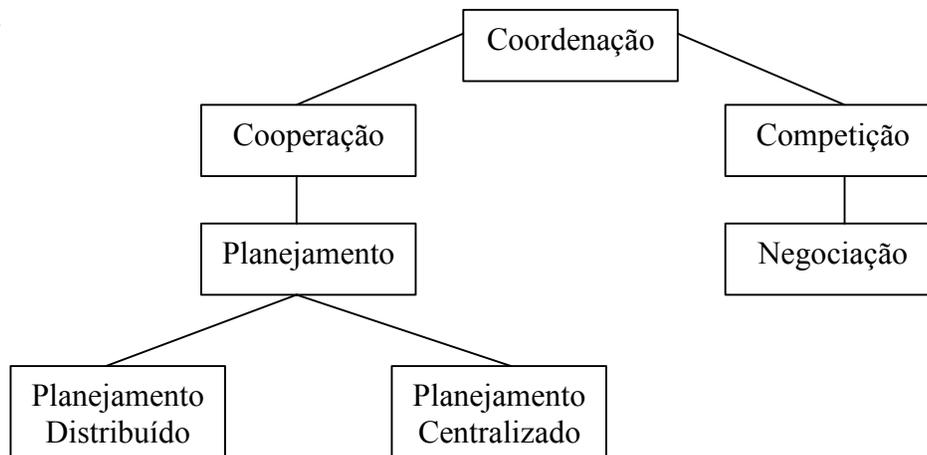


Figura 7.1: Taxonomia de como os agentes podem coordenar suas ações e comportamentos

A sociabilidade mencionada por Huhns e Stephens também é identificada por Durfee et al [DUR 89a] quando ele cita os três maiores pontos que devem estar presentes para uma coordenação ter sucesso: existir estruturas que permitam aos agentes interagir de maneira previsível; existir flexibilidade para que os agentes possam operar em ambientes dinâmicos e possam lidar com a sua visão imprecisa e parcial da comunidade; e os agentes devem ter conhecimento e capacidade de raciocínio suficiente para explorar estruturas e flexibilidade disponíveis.

A maioria das pesquisas em coordenação de agentes tem se concentrado em técnicas de distribuição de controle e dados [HUH 99] [CIA 2000]. Controle distribuído significa que os agentes têm um grau de autonomia na geração de novas ações e no processo de decisão de qual o próximo objetivo a ser atingido. A desvantagem da distribuição de controle e dados é que o estado completo do sistema está disperso pelo sistema e cada agente possui somente uma perspectiva parcial e imprecisa.

Embora o formalismo distribuído tenha sido usado freqüentemente para caracterizar problemas locais e globais, assim como Jennings [JEN 93c] [JEN 96], Huhns e Stephens [HUH 99] salientam que as estruturas chaves para os agentes envolvidos em um processo de coordenação são comprometimento e convenção. Se as circunstâncias não mudam, um agente irá honrar seus comprometimentos. A obrigação imposta pelo comprometimento restringe as futuras decisões do agente sobre a tomada de novos comprometimentos, pois ele sabe que um número suficiente de recursos deve ser reservado para honrar os comprometimentos existentes. Desta maneira, os comprometimentos dos agentes devem ser consistentes internamente e consistente com as suas crenças. Comprometimentos e convenções são as peças fundamentais para a coordenação: comprometimentos fornecem a estrutura necessária para interações previsíveis, e convenções sociais fornecem um grau necessário de suporte mútuo.

Como mencionado por Huhns and Stephens [HUH 99] a coordenação pode ser alcançada através de cooperação ou de competição. Nas próximas seções iremos apresentar alguns conceitos relacionados a estas duas abordagens.

7.2 Cooperação

Cooperação é freqüentemente apresentada como um dos conceitos principais que diferencia sistemas multiagentes de outras disciplinas relacionadas tais como: computação distribuída, sistemas orientados a objetos e sistemas especialistas. Embora seja considerada uma questão importante para sistemas de agentes, a cooperação, assim como vários conceitos dentro da inteligência artificial, não possui uma conceituação formal aceita por todos os pesquisadores da área.

Franklin aborda cooperação de uma maneira ampla, propondo uma tipologia (figura 7.2) e inserindo-a no contexto de sistemas multiagentes [DOR 96]. Um sistema multiagente é independente se cada agente possui a sua própria agenda [FRA 97] independentemente dos outros. Um sistema multiagente é discreto se ele é independente, e se as agendas dos agentes não suportam relação com outro agente. Sistemas discretos não envolvem cooperação. Contudo, os agentes podem cooperar mesmo que não tenham intenção de fazer isto. Cooperação emergente ocorre quando do ponto de vista do usuário os agentes estão trabalhando em conjunto, mas do ponto de vista dos agentes eles não estão.

O complemento dos sistemas independentes são sistemas nos quais as agendas dos agentes incluem cooperação com outros agentes do sistema (sistemas cooperativos). Tal cooperação pode ser comunicativa, pois os agentes se comunicam uns com os outros para cooperar ou pode ser não comunicativa. Neste caso, os agentes coordenam suas atividades cooperativas pela observação e reação ao comportamento do outro. A comunicação intencional pode ter pelo menos duas formas: agentes podem deliberar ou os agentes podem negociar. Nos sistemas deliberativos os agentes planejam suas ações em conjunto a fim de cooperar uns com os outros. Tal cooperação pode ou não incluir coordenação. Sistemas de negociação são como sistemas deliberativos, exceto que eles adicionam o fator de competição.

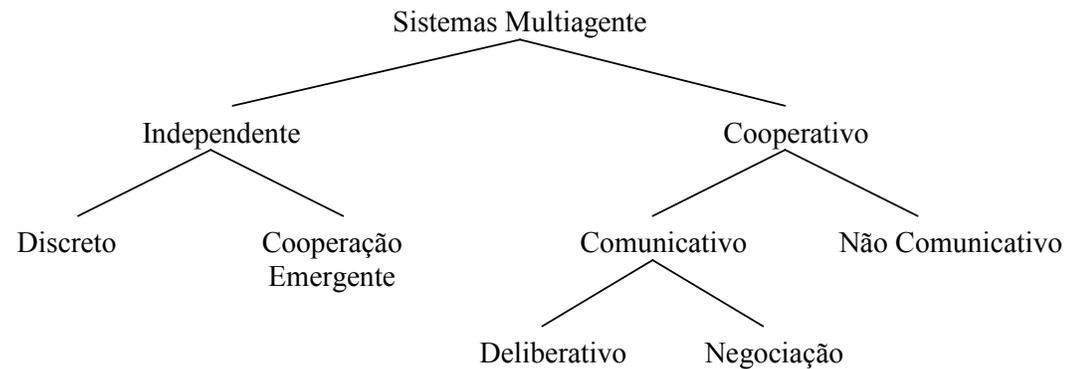


Figura 7.2: Tipologia de cooperação e seu enquadramento em sistemas multiagente

Doran [DOR 96] oferece um ponto de vista alternativo de cooperação. Para Doran a cooperação ocorre quando as ações dos agentes envolvidos satisfazem uma ou ambas condições:

1. os agentes têm um objetivo comum (possivelmente implícito, que nenhum agente pode alcançar de maneira isolada) e suas ações tendem a alcançar estes objetivos;
2. os agentes executam ações que possibilitam ou atingem não somente seus próprios objetivos, mas também os objetivos de outros agentes.

Esta definição se assemelha à cooperação emergente de Franklin, pois não é necessária a deliberação e os objetivos não precisam estar explícitos nos agentes. Contudo, a visão de cooperação inclui outros aspectos além desta visão behaviorista. Para os projetistas é importante saber o que leva a escolha de ações que resultam em cooperação. Doran identifica dois fatores importantes: tipo de processamento intra-agente que leva às ações cooperativas e princípios de decisão para escolher qual ação cooperativa seguir.

Norman [DOR 96] argumenta que cooperar é agir com outro ou outros para um propósito ou benefício comum. Mas o que significa agir para um propósito comum? O objetivo de um agente é que dirige o seu comportamento, mas existem duas maneiras primárias para dar um objetivo a um agente. O agente pode ter um conjunto de comportamentos que são projetados para que ele possua algum objetivo; em tal sistema de controle o objetivo é implícito no seu projeto (um sistema orientado a objetivos ou sistema de controle puramente baseado em comportamento). Uma alternativa é o agente ser

motivado por objetivos explícitos, possivelmente derivados de motivos mais básicos. Tal agente pode, então, aplicar planejamento e outros processos de tomada de decisão para dirigir suas ações em direção ao alcance destes objetivos (sistema de controle dirigido a objetivos). Contudo, compartilhar um mesmo objetivo não é suficiente para que os agentes cooperem, também é necessário que os agentes pretendam agir em conjunto (um comprometimento para uma tarefa conjunta). Isto vai contra as idéias de Franklin e Doran que não necessitam de uma intenção para cooperar.

7.2.1.1 *Mecanismos para Cooperação*

Uma estratégia básica utilizada por muitos mecanismos para cooperação é decompor e depois distribuir tarefas. Esta abordagem se baseia no método dividir-e-conquistar e pode reduzir a complexidade de uma tarefa, pois subtarefas menores necessitam de agentes com menor capacidade de processamento e poucos recursos. Contudo, o processo de decomposição deve considerar os recursos e capacidades dos agentes e decidir entre diferentes decomposições, se existirem alternativas.

Existem vários mecanismos utilizados para distribuição de tarefas. Iremos apresentar dois destes mecanismos: estrutura organizacional e planejamento multiagente. O mecanismo de estrutura organizacional foi escolhido, pois permite associar papéis aos agentes no momento da estruturação da sociedade e o mecanismo de planejamento multiagente foi escolhido porque queremos mostrar uma abordagem para uma das mudanças propostas por Hayes-Roth e Doyle [HAY 98b] de que o planejamento se torna improvisação.

7.2.1.1.1 Estrutura Organizacional

A estrutura organizacional define papéis, responsabilidades, preferências e padrões de controle para agentes dentro de uma sociedade cooperativa [DUR 99]. Quando um agente se junta a uma sociedade, ele assume um ou mais papéis e adquire os comprometerimentos relativos àqueles papéis. Os agentes se unem a uma sociedade de maneira autônoma, mas são restringidos pelos comprometerimentos dos papéis adotados, ou seja, a sociedade define o contexto social no qual os agentes interagem.

A estrutura organizacional é o cenário mais simples de coordenação porque define implicitamente as responsabilidades, capacidades, conectividade e fluxo de controle dos agentes. Ela fornece uma estrutura para atividade e interação através da definição de papéis, caminhos de comunicação e relacionamentos de autoridade, que pode ser implementado através de técnicas de coordenação do tipo mestre/escravo ou cliente/servidor.

7.2.1.1.2 Planejamento Multiagente

O planejamento multiagente tem como objetivo evitar inconsistência ou ações e interações conflitantes através da construção conjunta de um plano que detalha todas as ações e interações futuras necessárias para que os agentes atinjam seus objetivos, intercalando execução com mais planejamento e replanejamento. Existem dois tipos de planejamento multiagente [NWA 96]:

- planejamento multiagente centralizado;
- planejamento multiagente descentralizado.

No planejamento centralizado existe, geralmente, um agente coordenador que cria todos os planos parciais ou locais para os agentes individuais. O agente coordenador modifica os planos parciais e os combina dentro de um plano multiagente onde as inconsistências e interações conflitantes são eliminadas. Kim [KIM 94] salienta que o maior apelo à centralização está na sua simplicidade, pois como descrito, um centro autoritário determina o que deve ser feito e impõe isto sobre o restante do sistema. Contudo, Kim [KIM 94] considera o planejamento centralizado evasivo e embasa seu ponto de vista em uma limitação bastante forte do planejamento centralizado: como o controle está centralizado em um único agente, se ocorrer falha neste agente todo o sistema irá falhar.

Uma opção para superar esta limitação do planejamento centralizado é o planejamento descentralizado ou distribuído. No planejamento multiagente distribuído a idéia é fornecer para cada agente um modelo dos planos dos outros agentes. Os agentes se comunicam a fim de construir e atualizar seus planos individuais e seus modelos dos outros até que todos os conflitos sejam removidos.

O planejamento distribuído é um termo ambíguo, pois não deixa claro o que é distribuído e isto possibilita a existência de diferentes tipos de planejamento. Uma visão da distribuição é que o plano produzido pode ser distribuído, sendo que o plano resultante pode ou não ser distribuído. Neste caso, existem várias alternativas: planejamento centralizado para planos distribuídos, planejamento distribuído para planos centralizados e planejamento distribuído para planos distribuídos. A seguir vamos examinar resumidamente cada uma delas.

7.2.1.1.2.1 Planejamento Centralizado para Planos Distribuídos

O planejamento centralizado para planos distribuídos considera que planos que são executados de maneira distribuída podem ser formulados de maneira centralizada. Por exemplo, um planejador de ordem parcial pode gerar planos onde não precise existir uma ordem restrita entre algumas ações e onde as ações possam ser executadas em paralelo. Um agente coordenador centralizado com tal plano pode subdividir o plano em processos separados e estes são repassados aos agentes. Se os planos forem seguidos corretamente, e considerando a correção do conhecimento e previsibilidade do mundo, os agentes, operando em paralelo, alcançam um estado de mundo consistente com os objetivos do plano. O objetivo deste tipo de planejamento é encontrar, de todos os planos possíveis que atingem o objetivo, o plano que pode ser decomposto e distribuído mais efetivamente.

7.2.1.1.2.2 Planejamento Distribuído para Planos Centralizados

Em alguns casos, a construção de um plano completo pode necessitar da colaboração de diversos especialistas de planejamento cooperativos. Neste caso, cada um dos especialistas contribui para a construção de partes do plano até que um plano completo seja criado. Esta abordagem é chamada de compartilhamento de resultados, pois cada agente gera um plano parcial em paralelo e compartilha e combina estes planos para convergir em um plano completo usando técnicas de negociação.

7.2.1.1.2.3 Planejamento Distribuído para Planos Distribuídos

O planejamento distribuído para planos distribuídos ocorre quando o processo de planejamento e os seus resultados tendem a ser distribuídos. Neste caso, não é necessário ter um plano multiagente inteiramente representado no sistema, pois os planos distribuídos devem ser compatíveis. Esta compatibilidade pressupõe que os agentes não podem entrar em conflito uns com os outros quando estiverem executando seus planos e que os agentes devem ajudar os outros a atingir os seus planos quando isto for racional, ou seja, quando ao auxiliar outros agentes eles não estejam prejudicando a si próprios.

Existem várias técnicas para este tipo de planejamento, neste capítulo iremos apresentar resumidamente duas delas: combinação de planos (*plan merging*) e construção interativa de planos (*interactive plan formation*) [HUH 99].

Combinação de Planos

O método de combinação funciona da seguinte maneira: dados os planos de vários agentes (onde cada plano é assumido como um plano individual correto), o método começa pela análise das interações entre pares de ações a serem tomadas por diferentes agentes. Arbitrariamente, se considera as ações a_i e b_j como as próximas ações a serem executadas pelos agentes A e B. As ações a_i e b_j podem ser executadas em paralelo, se as condições e efeitos de cada uma das ações é satisfeita ao mesmo tempo com nenhuma das condições de outra ação. Se este é o caso, as ações podem comutar e são essencialmente independentes. Se este não é o caso, então ainda pode ser possível para as ações serem executadas em uma ordem restrita. Se a situação antes de qualquer ação ser tomada, modificada pelos efeitos de a_i , pode satisfazer as condições de b_j , então a_i pode preceder b_j . Também é possível para b_j preceder a_i . Caso não exista precedência possível entre as ações então existe conflito.

Formação Interativa de Planos

Uma alternativa à abordagem anterior é uma variação da abordagem de planejamento distribuído hierárquico. Esta variação permite que cada agente represente seu comportamento planejado localmente em múltiplos níveis de abstração, qualquer um deles podendo resolver todos os problemas. Nesta busca pelo espaço de comportamento hierárquico, primeiro se identifica o nível de abstração particular que deve ser trabalhado, e se os conflitos devem ser resolvidos neste nível ou passados para um nível mais detalhado. Depois, se utiliza uma busca de satisfação de restrições para resolver os conflitos. Como os planos em vários níveis de abstração ditam o comportamento dos agentes para um grau particular, esta abordagem tem sido caracterizada como uma busca através de um espaço de comportamentos hierárquico.

A vantagem de se resolver um conflito em um nível abstrato é que isto reduz a quantidade de busca, possibilitando que os planos sejam coordenados em menos tempo e com menos troca de mensagens. A desvantagem é que as restrições de coordenação no nível abstrato podem impor limites desnecessários em ações mais detalhadas. Em um nível mais detalhado, problemas precisos de interação podem ser reconhecidos e resolvidos, enquanto em um nível abstrato podem surgir soluções de coordenação mais ineficientes. As *tradeoffs* entre decisões de coordenação a longo-termo, simples, mas possivelmente

ineficientes versus decisões de coordenação mais responsável, mas complexas em tempo de execução são invariavelmente dependentes do domínio.

7.3 Competição

De acordo com a hierarquia apresentada por Huhns e Stephens [HUH 99] (figura 7.1), competição é o tipo de coordenação contrário à cooperação. Contudo, ao definir os conceitos propostos na hierarquia, Huhns e Stephens não definem competição e sim negociação que sob o nosso ponto de vista é um tipo de competição e não a competição em si. O mesmo fato ocorre em outros trabalhos estudados. Parece que existe uma generalização na área de que negociação é competição e vice e versa. Tomando esta premissa como verdadeira, não seria necessária a definição de um dos conceitos já que eles representam a mesma coisa. Não concordamos com esta premissa e mostraremos brevemente que ela não pode ser verdadeira devido ao próprio significado de cada palavra. Para embasar nossa afirmação recorreremos ao dicionário Aurélio [FER 99]. De acordo com o dicionário Aurélio:

- a palavra cooperação significa: ato ou efeito de cooperar e cooperar significa operar ou obrar simultaneamente; trabalhar em comum; colaborar, ajudar, auxiliar, colaborar;
- a palavra competição significa: ato ou efeito de competir; busca simultânea por dois ou mais indivíduos de uma vantagem, uma vitória, um prêmio; luta, desafio, disputa, rivalidade;
- a palavra competir significa: pretender uma coisa simultaneamente com outrem; rivalizar, emular; pertencer por direito; caber, tocar; ser da competência, cumprir, caber, impender ; rivalizar-se; emular-se;
- a palavra negociação significa: ato ou efeito de negociar; negócio; entendimento entre nações, por intermédio de seus legítimos representantes, para concluir tratados e convênios;
- a palavra negociar significa fazer negócio; exercer o comércio; comerciar; fazer negócio, negociar; manter relações para concluir tratados ou convênios; promover o andamento de; contratar, ajustar, agenciar; concluir, ajustar, celebrar; comprar ou vender, permutar, trocar e ajustar, combinar, concluir.

Segundo as definições do dicionário Aurélio competir e negociar não são sinônimos e competir e cooperar são opostos, pois competir é pretender alguma coisa simultaneamente com outrem e cooperar é operar ou obrar simultaneamente, trabalhar em conjunto.

Embora negociação não seja sinônimo para competição todos os trabalhos que dizem enfocar agentes competitivos definem somente o conceito de negociação. Desta maneira, na próxima seção apresentaremos alguns conceitos de negociação para diferentes autores.

7.3.1 Negociação

Embora negociação seja um aspecto importante para a modelagem de sistemas multiagentes, não existe uma definição clara e comum do que é negociação. Negociação é

utilizada mais especificamente para alocação de tarefas e recursos, reconhecimento de conflitos, resolução de disparidades de objetivos, determinação de estrutura organizacional e propicia à coerência da sociedade de agentes. A idéia compartilhada pelos diferentes pesquisadores que trabalham com negociação é de que os agentes usam negociação para resolução de conflitos e coordenação.

Alguns pesquisadores estudam negociação como um problema de linguagem, onde a sintaxe e a semântica da mensagem são analisadas. Outros exploram o processo de negociação e outros consideram os aspectos comportamentais dos negociadores. Como estes pontos de vista são diferentes fica claro porque não existe uma definição comum ou uma teoria formal sobre negociação [MÜL 96].

Um dos protocolos mais estudados para negociação utiliza a metáfora da organização humana. O protocolo de rede de contrato (PRC) tem sido uma das mais influentes abordagens de negociação proposta para SMA. Foi inspirado no processo de contratação nas organizações humanas. Os agentes coordenam suas atividades através de contratos para atingir objetivos específicos. Um agente agindo como gerente decompõe seu contrato (tarefa ou problema) em subcontratos a serem alcançados por outros agentes, potenciais contratados. Para cada subcontrato o gerente anuncia uma tarefa para a rede de agentes. Os agentes recebem e avaliam o anúncio. Agentes com recursos, especialidade e informação apropriada respondem ao gerente com ofertas que indicam sua habilidade para atingir a tarefa anunciada. O gerente avalia uma oferta recebida e encaminha a tarefa para o agente mais apropriado, chamado de contratado. O gerente e o contratado trocam informações durante o processo de resolução da tarefa [MOU 96]. Para Davis e Smith [DAR 88] negociação possui três componentes: (a) existe uma troca de informações em dois sentidos; (b) cada parte da negociação avalia a informação do seu ponto de vista e (c) o acordo final é atingido por seleção mútua.

Esta definição contém dois elementos principais para negociação: comunicação e tomada de decisão. O PRC desenvolvido por Davis e Smith é um protocolo fixo para troca de informações onde não existe possibilidade de barganha entre os agentes e um acordo mútuo é eventualmente dado pelo agente que toma a decisão da oferta. O PRC está mais para um método de coordenação padronizado do que para um princípio de negociação apesar de existir a idéia de que um agente negocia a realização de um serviço por parte de outro agente.

Outra definição é de que negociação é um processo pelo qual uma decisão conjunta é feita por uma ou mais partes. As partes primeiro verbalizam demandas contraditórias e então se movem em direção a acordos através de um processo de concessão ou busca por novas alternativas [MÜL 96] [SIE 97].

De acordo com [JEN 2001], as pesquisas em negociação abordam três grandes tópicos:

- protocolos de negociação: o conjunto de regras que governam a interação. Isto inclui os tipos de participantes, os estados de negociação, os eventos que causam mudanças nos estados de negociação e as ações válidas dos participantes em cada estado particular;
- objetos de negociação: as variedades de questões sobre as quais deve se chegar a um acordo. Em um extremo um objeto pode ser uma questão única (como um preço),

enquanto no outro extremo ele pode envolver várias questões (como preço, qualidade, tempo, condições, etc.).

- modelos de tomada de decisão dos agentes: o modelo de tomada de decisão tem como objetivo agir sobre o protocolo de negociação para atingir os objetivos de negociação. A sofisticação do modelo depende do protocolo de negociação, da natureza dos objetivos de negociação e das operações que podem ser executadas sobre os objetos.

Nesta abordagem a negociação pode ser vista como um espaço de busca distribuído através de um espaço de potenciais acordos. A dimensionalidade e topologia deste espaço é determinada pela estrutura do objeto de negociação. Para uma determinada negociação, os participantes são os componentes ativos que determinam a direção da busca. No início deste processo, cada agente tem uma porção do espaço no qual ele está apto a fazer acordos. Durante o processo de negociação, o espaço de acordo dos participantes pode mudar devido às mudanças que ocorrem no ambiente ou devido as suas mudanças de visão do problema.

Guttman e Maes [GUT 98a] [GUT 98b] definem negociação como uma forma de tomada de decisão onde duas ou mais partes procuram em conjunto um espaço de soluções possíveis com o objetivo de atingir um consenso. A partir desta definição eles apontam a existência de dois tipos de negociação: negociação competitiva e negociação cooperativa.

Negociação competitiva pode ser descrita como um processo de resolver um conflito envolvendo duas ou mais partes sobre um único objetivo mutuamente exclusivo. A literatura da teoria dos jogos descreve esta situação como um jogo de soma zero, um lado ganha e outro perde.

Negociação cooperativa pode ser descrita como um processo de tomada de decisão de resolver um conflito envolvendo duas ou mais partes sobre múltiplos objetivos interdependentes, mas não mutuamente exclusivos. O estudo de como analisar decisões multiobjetivos vem das pesquisas em economia e é chamado de teoria da utilidade multiatributo. A literatura da teoria dos jogos descreve negociação cooperativa como um jogo de soma não zero, onde os valores da múltiplas dimensões vão em diferentes direções, onde todas as partes saem ganhado. A negociação cooperativa é uma negociação do tipo ganho-ganho, ao contrário da negociação competitiva que é um tipo de negociação ganho-perda.

Segundo as definições apresentadas, os principais componentes da negociação são: comunicação, aspecto sem o qual a negociação não seria possível; decisão que é necessária para modelar o raciocínio de negociadores durante a negociação e modelo procedural e comportamento global dos negociadores durante todo o ciclo de negociação.

A existência de vários conceitos sobre cooperação e negociação gera uma confusão em relação à própria definição dos conceitos. Alguns autores vêem cooperação e negociação como conceitos diferentes [HUH 99], [DOR 97], [WOO 98], outros os misturam e falam, implicitamente ou explicitamente, sobre negociação cooperativa e competição negociada [GUT 98a] [GUT 98b], [ZHA 2000], [KRA 2001], [ORT 2000], [JEN 2001], [ZLO 96a] como mencionamos no parágrafo acima. Às vezes o mesmo autor em um momento considera os dois conceitos separadamente e em outro momento como conceitos combinados. A fim de esclarecer estas diferentes interpretações propomos uma

nomenclatura sistemática e extensível chamada de espaço de mecanismos de coordenação. Inicialmente especificamos o espaço composto por duas dimensões: estilos de coordenação e técnicas de coordenação. O espaço das formas de coordenação introduz uma nova técnica de se atingir coordenação, através da improvisação (apresentada no capítulo 4).

7.4 Espaço de Formas de Coordenação

Entendemos que a hierarquia apresentada por Huhns e Stephens [HUH 99] divide os tipos de mecanismos de coordenação em estilos de coordenação (cooperação e competição) e técnicas (planejamento e improvisação) para se atingir estes estilos. Desta maneira, propomos a separação dos estilos e técnicas presentes na hierarquia em duas dimensões diferentes, a fim de criar o espaço de estilos e técnicas, onde podemos classificar os diferentes trabalhos desenvolvidos em coordenação de agentes.

7.4.1 Eixo dos Estilos

Assumimos que cooperação e competição são dois estilos opostos possíveis de coordenação e cada um está localizado em um extremo do eixo dos estilos. Desta maneira, os extremos já estão definidos, mas quais são as posições intermediárias entre eles? Como podemos entender estas posições intermediárias? Propomos entender estas posições intermediárias usando uma hierarquia de níveis composta por três categorias. Estas categorias são inter-relacionadas e cada nível tem características adicionais em relação aos seus precedentes:

- 1º nível - nível de objetivos: é o nível mais baixo, onde a classificação ocorre através dos tipos de objetivos dos agentes. Os agentes neste nível são classificados como cooperativos ou competitivos. Agentes cooperativos são aqueles que compartilham os mesmos objetivos, todos trabalham em conjunto para tentar solucionar um objetivo global compartilhado por todos. Agentes competitivos são aqueles que possuem objetivos distintos e que enfocam a resolução dos seus objetivos pessoais.
- 2º nível – nível de acordos: à medida que os agentes tentam atingir seus objetivos, sejam eles compartilhados ou não, podem surgir conflitos e estes conflitos podem ser solucionados através de acordos. A existência ou não de acordos e o seu tipo (mutuamente ou exclusivante benéfico) formam o nível intermediário de classificação. O surgimento de um conflito está vinculado à existência ou não de recursos para realização de suas tarefas.
- 3º nível – nível de comportamento: este é o nível mais alto da hierarquia e está relacionado à maneira como os agentes se comportam para atingir seus objetivos, considerada a presença ou não de conflitos. Entendemos que existem dois extremos que influenciam no comportamento do agente, a capacidade de escolher livremente como agir e a existência de uma imposição externa que restrinja a sua capacidade de ação. O primeiro caso, ou seja, a livre escolha, está relacionada à cooperação e o segundo caso, a imposição, está relacionado ao extremo da competição.

Cada agente possui seu objetivo, seja este comum ou distinto em relação aos outros agentes, e este objetivo é constituído por uma hierarquia de objetivos que vai sendo montada à medida que o agente tenta atingir seu objetivo. O objetivo do agente é tido como

um objetivo fim que pode ou não ser subdividido em objetivos intermediários que são os meios de se alcançar o fim. Os objetivos intermediários são criados, quando o agente não possui recursos suficientes para atingir seu objetivo fim. Enquanto o agente não possui recursos para executar seu objetivo meio, ele o transforma em um novo objetivo fim e assim vai criando a hierarquia de objetivos-subobjetivos. A construção desta hierarquia de objetivos-subobjetivos pode ser entendida como um processo de resolução de conflito que pode envolver acordos mais, menos ou igualmente benéficos para os agentes envolvidos. Os graus destes acordos serão variáveis, considerando-se a capacidade de livre escolha e imposição que os agentes possuem. A imposição neste caso pode ser entendida como restrições ou impedimentos relacionados ao agente ou ao ambiente em que ele está inserido. Podemos dizer que a falta de recursos é um fator que influencia nos graus de livre escolha e imposição.

Negociação é um mecanismo que se encontra dentro do eixo de estilos. Quando existe negociação sempre existe um acordo entre as partes envolvidas. Dependendo dos benefícios que estes acordos trazem aos agentes envolvidos na negociação e dependendo de como eles agem para resolver estes acordos, os agentes vão se encontrar em um dos estilos, ou entre os estilos: cooperação negociada, cooperação e competição balanceada e competição negociada. Na cooperação pura e na competição pura (sem negociação) os agentes nunca negociam o uso de recursos, ou porque nunca ocorre um conflito ou porque nunca ocorre um acordo. A figura 7.3 mostra as categorias presentes no eixo dos estilos.

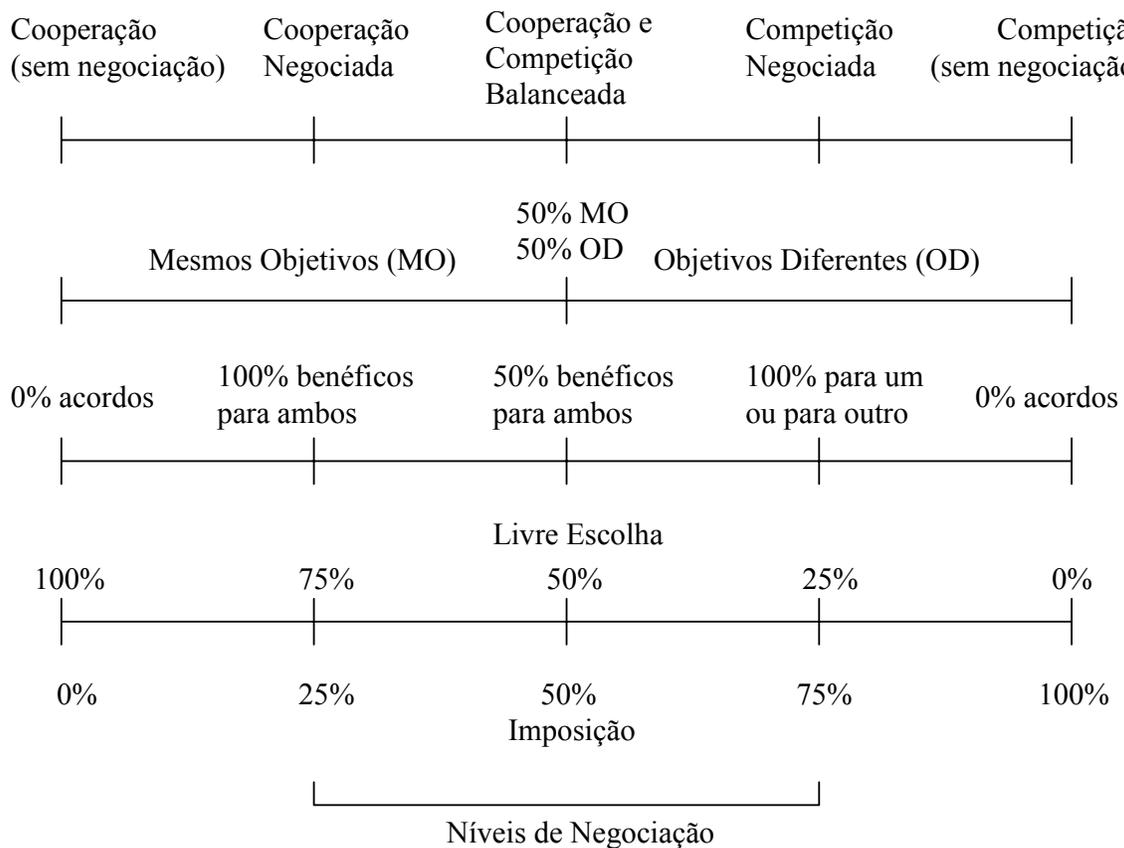


Figura 7.3: Categorias do Eixo dos Estilos

7.4.2 Eixo das Técnicas

Analisando a hierarquia apresentada por Huhns e Stephens [HUH 99], podemos observar que as técnicas ditam a maneira como o agente atinge seus estilos, ou seja, ditam o comportamento do agente. Entendemos que essa hierarquia não representa uma variedade total de comportamentos possíveis, pois enfatiza o comportamento planejado com antecedência. Desta maneira, propomos as seguintes categorias para o comportamento do agente:

- improvisação: um dos extremos seria a execução de comportamento sem planejamento prévio, usando os recursos disponíveis de acordo com a situação do momento. Improvisação neste caso significa o que chamamos de “improvisação pura”. “Improvisação pura” é diferente de “reatividade pura”, porque o último atribui para o ambiente a responsabilidade pelo controle do comportamento, enquanto na improvisação o controle está com o agente.
- planejamento: outro extremo é o planejamento. Planejamento neste caso não significa o planejamento clássico da Inteligência Artificial como uma busca em um espaço de estados. Significa que o comportamento do agente será ditado por uma estrutura previamente construída, que considera as situações que podem ocorrer no futuro. Pode ou não incluir mecanismos presentes no planejamento da IA clássica, como pode também incluir outros mecanismos como teoria dos jogos, redes de contrato, etc.
- imp-plan: categoria intermediária que possibilita ao agente possuir tanto capacidade de improvisação quanto de planejamento e as duas em mesma quantidade.
- improvisação planejada: categoria que possui predomínio da improvisação, mas permite ao agente possuir alguma capacidade de planejamento.
- planejamento improvisado: categoria que possui predomínio do planejamento, mas permite ao agente possuir alguma capacidade de improvisação.

Através destas categorias montamos o eixo das técnicas, que é um espectro cujos comportamentos variam desde a improvisação pura até planejamento puro (figura 7.4).

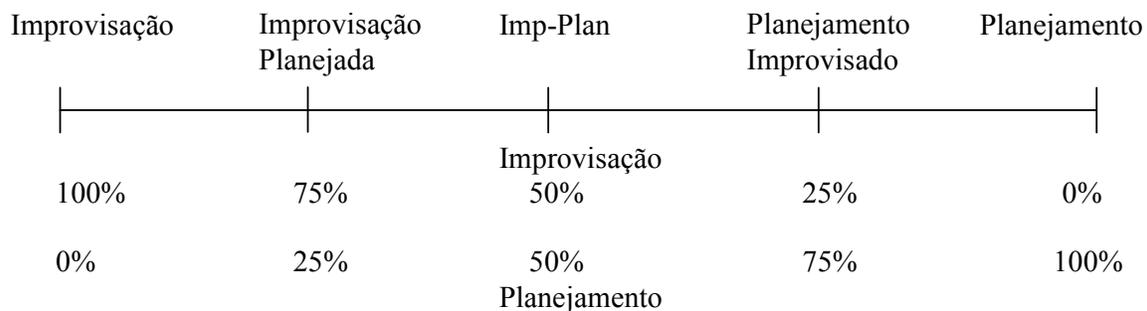


Figura 7.4: Categorias das Técnicas

7.5 Visão Geral das Formas de Coordenação e sua Classificação no Espaço de Coordenação

Nesta seção, iremos apresentar resumidamente alguns trabalhos envolvidos em coordenação e classificar as formas de coordenação por eles utilizados no espaço de formas de coordenação proposto na seção 7.4.

7.5.1 Huang e Sycara [HUA 2002]

O objetivo deste trabalho é construir modelos de negociação para agentes heterogêneos e de interesse próprio. Os autores modelam o processo de negociação como um processo de tomada de decisão seqüencial: em cada iteração da negociação, um agente checa o histórico do processo, atualiza suas crenças sobre os seus oponentes e então tenta maximizar seu benefício esperado baseado nas suas próprias crenças subjetivas. Esta abordagem modela os estados mentais dos agentes e examina seu impacto sobre o resultado da negociação. O modelo proposto por Huang e Sycara está relacionado à linha de pesquisa em crenças, desejos e intenções, no qual os agentes se adaptam às incertezas do ambiente, usando diferentes políticas de reconsideração de intenções.

A negociação é modelada como um jogo com M estados jogados por N jogadores em um tempo limitado. A estrutura do jogo é definida como um autômato de transição de estados, e os jogadores sempre irão selecionar e executar a ação que maximiza o seu benefício esperado. Neste modelo, as ações dos agentes são sempre observadas. Desta maneira, os jogadores notam as ações dos seus oponentes e as analisam com base em crenças subjetivas. Cada crença subjetiva do jogador é adquirida através da sua própria experiência e não é compartilhada com outros jogadores. As crenças são utilizadas no momento de selecionar uma ação para ser executada. Deste modo, as crenças do jogador irão determinar seu comportamento durante a negociação. As crenças são fixas durante cada iteração. O jogador somente pode atualizar suas crenças entre uma iteração e a próxima.

Classificamos esta forma de coordenação como competição negociada e planejamento improvisado. Entendemos que é competição negociada, pois os agentes possuem objetivos distintos, existe conflito para obtenção de recursos, sempre existindo um acordo que é melhor para um agente do que para outro, e os agentes possuem alguma capacidade de livre escolha para resolução dos seus conflitos. Entendemos que é planejamento improvisado, pois a forma possui algo de improvisação, visto que os jogadores (agentes) podem escolher, entre vários estados, aquele que é mais apropriado, considerando as crenças e função de utilidade do jogador. Contudo, esta improvisação é restrita, porque os jogadores somente podem atualizar suas crenças entre as iterações e não durante uma iteração. O último caso seria o mais apropriado e permitiria aos agentes se adaptar à situação atual, isto é improvisar.

7.5.2 Kraus [KRA 2001]

Kraus se preocupa com a cooperação e coordenação de sistemas inteligentes (agentes) que são de interesse próprio e que são provavelmente propriedade de diferentes organizações ou indivíduos. Desta maneira, sempre surgem conflitos entre os agentes, sendo necessária a utilização de um mecanismo que chegue a um acordo entre eles, um

destes mecanismos é a negociação. Kraus [KRA 2001] apresenta um modelo de estratégia de negociação que pode ser usado para o desenvolvimento de agentes autônomos que atingem acordos mutuamente benéficos de forma eficiente em ambientes complexos. Este modelo integra teoria dos jogos, técnicas econômicas e métodos heurísticos da inteligência artificial.

Classificamos a forma de coordenação utilizada no trabalho de Kraus como competição negociada e num nível intermediário entre planejamento e improvisação. Embora não tenha sido especificado claramente, a noção de função de utilidade e estratégia de negociação parece possibilitar alternativas à execução das ações dos agentes. Contudo, estas alternativas são dirigidas, pois tanto as funções de utilidade quanto a estratégia são impostas pelo projetista. O projetista é responsável por setar as funções de utilidade e estratégias, e os agentes não podem modificá-las. Desta maneira, a técnica mais adequada é o planejamento improvisado.

7.5.3 Ortiz et al [ORT 2000]

O enfoque do trabalho é no problema de alocação de recursos e tarefas distribuídas para domínios dinâmicos. O domínio estudado envolve a coalizão de agentes cooperativos que devem negociar a distribuição de tarefas. Para resolver este problema, os autores apresentam uma rede de comprometimento limitada a tempo. Esta rede de comprometimento é um mecanismo de negociação híbrido, baseado em extensões dos leilões de tarefas e redes de contrato que suportam a noção de comprometimento recíproco.

O mecanismo de negociação funciona da seguinte maneira: os times negociam a distribuição de novas tarefas iniciando os leilões para potenciais membros dos times que então dizem na oferta o quão bem eles podem executar aquelas tarefas. Como o mecanismo é baseado em redes de contrato, podem existir subleilões, e a sua profundidade pode variar dependendo do tempo disponível para realização dos mesmos. Contudo, ao contrário das redes de contrato onde o comprometimento do contratado para o contratante é *one-side*, as redes de comprometimento permitem reciprocidade entre o contratado e outro membro do grupo: se o contratado pode identificar uma tarefa que poderia auxiliar outro membro do grupo, então ele procura fazer isto. Além disso, as redes de comprometimento usam a idéia presente na alocação combinatória de tarefas: conjuntos de tarefas podem ser anunciados em qualquer momento e os licitadores podem fazer ofertas em qualquer subconjunto destas tarefas.

As redes de comprometimento introduzem três construtores não presentes nas redes de contrato padrão: *persistent bids* (ofertas podem persistir por padrão, ou seja, em situações críticas se tem a vantagem de que se uma nova tarefa aparece em uma localização próxima à coberta pela oferta recente, pode-se simplesmente atribuir esta oferta ao licenciado correspondente. Os agentes podem anunciar que a oferta padrão não é mais válida devido a algum novo comprometimento), *reciprocal background commitments* (agentes devem demonstrar possibilidade de fornecer comportamento de ajuda para outros membros do grupo), *contingent commitments* (agentes podem largar comprometimento se as condições mudam de modo a garantir a ação).

Classificamos o trabalho de Ortiz et al [ORT 2000] como cooperação negociada, porque os agentes são cooperativos, ou seja, possuem objetivos comuns e utilizam um mecanismo de negociação baseado em rede de contrato que fornece acordos mutuamente

benéficos para ambas as partes. A técnica utilizada pelo trabalho para atingir a cooperação negociada é o planejamento, pois o modelo não apresenta nenhuma possibilidade de alternativas e improvisação em relação à situação atual para os agentes. É um modelo baseado em modificações da rede de contrato, ou seja, um modelo que apresenta uma maneira bem específica de como agir não dando margem à improvisação do comportamento do agente.

7.5.4 Guttman e Maes [GUT 98a] [GUT 98b]

Nos trabalhos de Guttman e Maes sobre negociação em sistemas multiagentes, são propostos dois tipos de negociação: negociação distributiva e negociação integrativa. A negociação distributiva foi aplicada no *Kasbah* [CHA 96] [GUT 98a] [GUT 98b]. *Kasbah* é um sistema multiagente baseado na *Web* onde os usuários criam agentes vendedores e compradores para auxiliar na transação de bens. O usuário seta parâmetros para os agentes compradores e vendedores, e estes parâmetros irão definir os objetivos dos agentes. O usuário pode especificar a estratégia de negociação através da função que os agentes usam para baixar o seu preço, à medida que o prazo de venda se aproxima. Os usuários também definem parâmetros para guiar a estratégia de negociação dos agentes compradores.

Entendemos que o estilo de cooperação do *Kasbah* é a competição negociada, e a técnica utilizada no trabalho para atingir este estilo é o planejamento improvisado. É competição negociada, porque os agentes possuem objetivos distintos, como os próprios autores salientam, mutuamente exclusivos, o que ocasiona conflitos que são resolvidos por meio de acordos que são melhores para um agente do que para outro. Vai estar entre planejamento e improvisação, pois os agentes possuem uma função para reduzir ou aumentar o preço do bem, se eles não conseguem chegar a um acordo. Contudo, esta improvisação é bastante limitada, pois a função de decaimento ou aumento é fornecida pelo usuário e não pode ser modificada durante uma interação. A negociação ocorre no mercado, e é o mercado que especifica o protocolo de negociação. A estratégia de negociação está dentro de cada agente, através da suas funções para maximizar ou minimizar o preço do produto, dependendo se está sendo feita uma venda ou uma compra. O protocolo é representado por mensagens que irão especificar o que está sendo oferecido, qual o preço, qual o item, adicionar potenciais consumidores, adicionar potenciais vendedores, agente sai do mercado, agente fecha negócio. O protocolo pode ser visto como um tipo de rede de contrato modificado, onde existe a presença de um agente centralizador que coordena as interações iniciais dos agentes compradores e vendedores.

A negociação integrativa foi aplicada no sistema *Tete-a-Tete*. *Tete-a-Tete* utiliza teoria de utilidade multiatributo (TUMA) e problemas de satisfação de restrições distribuídas (PSRD) para cooperativamente negociar através de múltiplos termos da transação [GUT 98b]. Desta maneira, o sistema *Tete-a-Tete* se enquadra entre cooperação e competição balanceada e competição negociada na área chamada de competição negociada cooperativa, pois trata objetivos distintos, mas não mutuamente exclusivos. Neste sentido, o tipo de acordos em relação à competição de recursos é quase meio a meio benéfico para ambos agentes envolvidos no processo, e os agentes possuem um grau de livre escolha maior do que os que se enquadram na competição negociada. Em relação ao eixo das técnicas, o trabalho está mais próximo do modo Imp-Plan. Entendemos que a teoria de utilidade multiatributo e o protocolo de satisfação de restrições distribuídas oferecem aos agentes alternativas de ações a serem executadas, contudo não é especificado se estas

restrições são informadas pelo projetista do sistema ou se são obtidas dinamicamente com o retorno do ambiente em relação à situação atual.

7.5.5 Wurman, Wellman e Walsh [WUR 98]

Na mesma linha do *Kasbah* está o trabalho *AuctionBot*. *AuctionBot* é uma plataforma para negociação baseada em preço. Um mecanismo de negociação é essencialmente um protocolo dentro do qual agentes interagem para determinar um contrato. Leilões constituem uma classe geral destes protocolos. Um leilão é uma instituição de mercado com um conjunto de regras explícitas que determina a alocação de recursos e preços com base nas ofertas dos participantes do mercado. As três atividades centrais de um leilão são: receber ofertas, fornecer informações intermediárias, esclarecer o propósito do leilão.

Assim como *Kasbah*, *AuctionBot* também se enquadra em competição negociada, pois os agentes têm objetivos distintos e mutuamente exclusivos e procuram por um acordo que sempre favorece seus próprios propósitos. O protocolo de negociação envolve o leilão de receber ofertas e aceitar ou rejeitar ofertas, executado através de trocas de mensagens. Como o protocolo de negociação é baseado em uma estrutura fixa e previamente definida, o *AuctionBot* é classificado dentro da técnica de planejamento.

7.5.6 Grupo de Lesser [SAN 95] [JEN 99] [ZHA 2000] [ZHA 2001]

Trabalhos desenvolvidos pelo grupo de Lesser enquadram negociação em sistemas multiagentes em duas principais categorias: negociação competitiva e negociação cooperativa. Lesser e seu grupo abordam negociação competitiva como uma extensão do protocolo de redes de contrato permitindo que agentes de interesse próprio façam acordos de maneira inteligente com a incerteza presente no processo de negociação [SAN 95]. A comunicação entre o contratante e o contratado é realizada através da troca de mensagens que especifica as alternativas envolvidas no acordo. Cada acordo tem um comprometimento relacionado e quando os agentes concordam com um acordo ambos assumem os comprometimentos relacionados. Se um agente quer quebrar um acordo ele tem que pagar por isto. O contratado pode enviar uma contraproposta para o contratante e este pode aceitá-la, rejeitá-la ou terminar a negociação.

O trabalho desenvolvido em 1995 pode ser categorizado como competição negociada, pois os agentes são de interesse próprio e sempre procuram atingir seus objetivos individuais. Isto ocasiona a existência de conflito, solucionado através de acordos que são benéficos somente para uma das partes envolvidas. Como o protocolo de negociação é uma extensão do protocolo da rede de contrato, podemos dizer que a técnica para se atingir a competição negociada é o planejamento, pois todas as mensagens e ações relacionadas aos agentes são especificadas antecipadamente, e os agentes não têm espaço para improvisação.

Na negociação cooperativa cada agente tenta alcançar a máxima utilidade global que considera a utilidade das atividades de todos os agentes e é usada no domínio de alocação de tarefas. O mecanismo de negociação cooperativa assume que existe a presença de um escalonador local que decide quais métodos executar a fim de maximizar a função de utilidade multidimensional.

O protocolo de negociação é descrito através de um modelo de máquina de estados finitos. Este protocolo é estruturado em três funções. A primeira é gerar a proposta inicial, a segunda é gerar uma contraproposta por parte do contratado e a terceira é para o contratante gerar uma nova proposta em resposta a contraproposta anterior. Usando estas funções os agentes tentam encontrar uma solução que maximize suas utilidades.

O trabalho desenvolvido em negociação cooperativa encontra-se no estilo competição negociada. A negociação cooperativa de Lesser assume que existe um mecanismo de escalonamento que decide qual método de execução de ações deveria ser realizado e quando. A técnica deste trabalho se enquadra no planejamento.

7.5.7 Zlotkin e Rosenchein [ZLO 89] [ZLO 90] [ZLO 96a] [ZLO 96b] [ROS 97]

Os trabalhos de Zlotkin e Rosenchein enfocam um tipo de negociação que ocorre entre agentes que servem interesses de partes distintas. Os agentes são autônomos no sentido de que possuem suas próprias funções de utilidade. A negociação neste cenário pode ser usada para compartilhar o trabalho associado com a execução de um plano conjunto ou para resolver conflitos que surjam devido à limitação dos recursos.

A idéia é usar ferramentas da teoria dos jogos para projetar ambientes de negociação e interação apropriadas para agentes que são programados por diferentes entidades e que buscam atingir diferentes objetivos. A abordagem desenvolvida foi estabelecer protocolos mutuamente aceitáveis para que os agentes os utilizassem com o objetivo de chegar a acordos. Zlotkin e Rosenchein apontam que estes protocolos são dependentes das características do domínio, e que uma classe de domínio que faz com que os agentes ajam de uma determinada maneira, não irá ser apropriada em outro tipo de domínio.

Desta maneira, Zlotkin e Rosenchein apresentam as classes de domínio em uma hierarquia de três níveis, onde cada nível é mais geral do que o seu antecessor. No domínio orientado a tarefa (DOT), a atividade do agente é definida em termos de um conjunto de tarefas que ele deve atingir. As tarefas podem ser realizadas sem se preocupar com a interferência de outros agentes. Todos os recursos necessários para se atingir a tarefa estão disponíveis no agente. A negociação tem como objetivo descobrir uma redistribuição de tarefas que seja mutuamente benéfica. Um exemplo é o domínio do entregador de cartas, onde dois agentes chegam no correio de manhã cedo e recebem os sacos de cartas que eles têm que entregar pela cidade e ambos têm que ir a um mesmo local. Os agentes cooperam antes de iniciar sua jornada no sentido de que um tenta convencer o outro de que já que ele está indo de qualquer maneira para aquele lugar ele pode entregar a carta para o outro. Este exemplo mostra que existem diferentes divisões das tarefas, algumas podem ser melhores para uns e outras para outros. Os agentes devem chegar a um acordo de como vão dividir as tarefas.

A maioria dos domínios clássicos da IA tem sido instanciadas para domínios orientados a estados (DOE). O principal atributo dos DOE é que os agentes podem ter efeitos colaterais. Nos DOT, não existem efeitos colaterais e geralmente todos os recursos são irrestritos. Logo, quando um agente atinge seu próprio conjunto de tarefas no DOT ele não gera nenhum efeito positivo ou negativo nos outros agentes. Ele não impede outro agente de atingir seu objetivo nem satisfaz os objetivos do outro agente “por acidente”. Contudo, no DOE onde existem efeitos colaterais, os agentes podem não intencionalmente atingir os objetivos do outro e ter benefício das ações dos outros agentes. O efeito é que

podem surgir interações negativas. Desta maneira, DOE é um domínio que não é necessariamente cooperativo, por causa dos efeitos colaterais das ações. Nos DOE, os agentes têm que tratar com conflito de objetivos e interferência, bem como a possibilidade de uma negociação não intencional.

Nos DOT o custo de um plano coordenado nunca pode ser pior do que a execução individual do plano pelo agente. No pior caso, cada agente executa o seu próprio conjunto de tarefas. No DOE isto não é verdade, pois podem existir interações negativas, ou interações que excluem condições, que significa que a execução da ação por parte do agente tem como efeito a não satisfação da condição de outro agente. Nos DOE pode também ocorrer que o objetivo de outro agente seja acidentalmente atingido ou pode ocorrer uma interação que habilita condições, ou seja, a execução de uma ação do agente tem como efeito a satisfação da condição da ação de outro agente.

Domínio orientado a custo (DOC) assume, como o DOE, que os objetivos especificam estados finais, mas este fato está codificado em uma função que rateia a aceitação de estados. Cada estado no mundo é melhor ou pior, mas não é a mesma noção binária do DOE. Em um DOC existe uma função de decisão teórica que o agente usa para procurar por estados melhores. O objeto de negociação é um plano conjunto, escalonamentos e relaxamento de objetivos. Os agentes podem não ser capazes de chegar a um estado final, mas eles podem chegar a um estado que é um pouco menos pior.

Zlotkin e Rosenchein propuseram um protocolo de negociação unificado que resolve resolução de conflitos e acordos cooperativos. O protocolo especifica funções de utilidade e outros conceitos relacionados a cada um dos domínios apresentados acima.

A partir das explicações apresentadas sobre os três diferentes tipos de domínios, podemos classificá-los dentro do nosso espaço de formas de coordenação. Entendemos que o domínio orientado a tarefas pertence à competição, e mais especificamente está muito próximo da categoria cooperação e competição balanceada, pois os agentes sempre chegam a acordos mutuamente benéficos. Os domínios orientados a estados não são necessariamente cooperativos, porque as ações têm efeitos, ou seja, a negociação é competitiva com possibilidade de ser cooperativa. Neste caso seria mais adequado enquadrá-lo entre competição negociada e competição negociada cooperativa. Este tipo de domínio produz os chamados acordos semicooperativos, pois em algum momento ocorre o conflito. O domínio orientado a custo é semelhante ao domínio orientado a estados. No domínio orientado a custo mesmo que os agentes não atinjam seus objetivos, eles tentam atingir um estado mais próximo possível do final. Deste modo, o domínio encontra-se um pouco mais para o lado da competição negociada cooperativa do que o domínio orientado a estados. Como todos os domínios utilizam um protocolo de negociação baseado na abordagem tradicional da IA de busca de espaços de estados, mas com uma função de utilidade particular para cada domínio, consideramos que os trabalhos encontram-se muito próximos do planejamento improvisado.

7.5.8 Jennings et al [JEN 2000] [JEN 2001]

Jennings et al [JEN 2000] trabalharam em três tipos diferentes de modelos de coordenação:

7.5.8.1 Modelos das Teorias de Jogos [VUL 2000]

Nesta linha de trabalho, os agentes são autônomos e geralmente representam diferentes organizações ou indivíduos. Este fato implica que os agentes têm diferentes objetivos e estão envolvidos em uma negociação competitiva. Vulkan e Jennings [VUL 2000] mostraram como para muitas negociações podem ser setadas como leilões Americanos para objetos de negociação com múltiplas dimensões. Em outras palavras, a negociação leva em conta um número de questões quantitativas (preço, duração e custo) e qualitativas (natureza do contrato). Para uma negociação ter sucesso, ela precisa resolver uma série de questões quantitativas e qualitativas. Estas questões representam as preferências dos agentes e são representadas como uma função de utilidade com diferentes pesos para preferências.

A partir desta breve explicação enquadraremos os modelos de negociação baseados na teoria de jogos, como competição negociada e entre planejamento e improvisação, mas com predominância no planejamento. O modo não está no seu extremo completo de planejamento, porque existe uma função de utilidade que considera as preferências dos agentes. Estas preferências possuem pesos associados que podem ser modificados. Logo o modo mais adequado é o planejamento improvisado.

7.5.8.2 Abordagens Heurísticas [FAR 98]

As abordagens heurísticas são usadas quando o projetista não pode impor *a priori* uma estratégia de negociação para os agentes. Desta maneira, Faratin et al [FAR 98] desenvolveram algoritmos de negociação baseados na teoria de utilidade multiatributo. Nestes algoritmos, os agentes têm uma estratégia de negociação que diz como negociar em um determinado conflito e uma família de táticas de negociação que formam a estratégia. O objetivo da estratégia de negociação é determinar o melhor curso de ação, que irá resultar em um acordo que maximiza a função de utilidade do agente.

Neste caso, assim como na teoria de jogos, os agentes possuem seus próprios objetivos que são diferentes dos outros agentes, o que ocasiona conflitos e acordos benéficos para uma das partes, por isso categorizamos as abordagens heurísticas como competição negociada. Como a estratégia de negociação é determinar o melhor curso de ação que irá resultar em um contrato que maximiza sua função de utilidade, a abordagem está mais para o modo de planejamento. Contudo, diferentes táticas podem formar uma estratégia, existindo a possibilidade de alternativas para os agentes, mas alternativas restritas às táticas pré-estabelecidas. Desta maneira, a técnica mais adequada é o planejamento improvisado.

7.5.8.3 Abordagens Baseadas na Argumentação [PAR 98a] [PAR 98b]

Em alguns casos poderia ser interessante que os agentes pudessem explicar e argumentar porque o outro agente tem que concordar com o acordo que é oferecido. Desta maneira, Parsons e Jennings [PAR 98a] e Parsons et al [PAR 98b] apresentam uma ferramenta baseada em um sistema de argumentação que permite aos agentes negociar, a fim de estabelecer maneiras aceitáveis de se resolver um problema. Esta ferramenta usa argumentação tanto no nível de raciocínio interno do agente quanto no nível de negociação entre os agentes. O protocolo de negociação tem a forma de um diagrama de transição de

estados que indica os vários estados possíveis que um agente pode ir durante uma negociação e as transições válidas entre estados.

Neste caso, os agentes também participam de uma competição negociada, pois os agentes possuem objetivos conflitantes. Como o protocolo de negociação é descrito como um diagrama de transição de estados, é uma coisa previamente planejada, as opções já foram escolhidas, entretanto cada uma das opções do diagrama pode representar uma alternativa que o agente vai escolher ao longo da execução. Desta maneira, entendemos que existe algo de improvisação, mas com ênfase no planejamento, logo a técnica mais adequada é o planejamento improvisado.

7.5.8.4 *Oliveira e Rocha [OLI 2001]*

Oliveira e Rocha [OLI 2001] desenvolveram um ambiente para gerar e explorar uma comunidade de agentes cooperativos. A estrutura básica dos agentes consiste de dois subsistemas: uma camada superior para cooperação e coordenação da atividade dos agentes e o sistema inteligente que tem as capacidades de um resolvidor de problemas específico. A camada de cooperação tem a responsabilidade de coordenar as atividades inter e intra-agentes e possui os seguintes módulos: comunicação (troca de informações com outros agentes), tomada de decisão (escolhe políticas de cooperação de acordo com a carga de trabalho do agente e controla o módulo de comunicação) e controle (escalona as atividades de resolução de problemas).

Para cooperar com outros agentes o módulo de tomada de decisão usa informação sobre o conhecimento e carga de trabalho dos outros agentes, informações sobre os seus recursos, um conjunto de regras para cooperação e de controle local e um quadro negro de dados e outro de objetivos. O monitor é o módulo da camada de cooperação que controla e monitora o sistema inteligente. Os agentes cooperativos devem garantir o escalonamento das tarefas locais, ou seja, a realização da cada tarefa que conta na agenda de acordo com o seu prazo. Quando um agente detecta a necessidade de ajuda, pois ele é incapaz de realizar uma tarefa, ele irá procurar se existem agentes capazes de realizar a tarefa em questão. Se existe um ou mais agentes, o primeiro passo é iniciar o processo de negociação. Um processo de negociação é executado através de um protocolo de negociação que permite a troca de informação entre os agentes. Este protocolo é composto pelo envio e recebimento de mensagens que tornam possível um acordo entre os agentes.

O trabalho pode se enquadrar em cooperação, mas cooperação que possui negociação. Está baseado em planejamento e não improvisação porque os agentes somente vão chegar a um acordo ou não através de um protocolo de trocas de mensagens. Os agentes possuem objetivos comuns, e quando existe a necessidade de compartilhar recursos, os acordos são mutuamente benéficos para ambos. Além disso, os agentes possuem maior capacidade de livre escolha do que a imposição ocasionada pelo ambiente ou por outros agentes. Logo, a categorização mais adequada é a cooperação negociada.

7.5.9 **Bonarini e Restelli [BON 2002]**

Bonarini e Restelli desenvolveram uma arquitetura multiagente chamada SCARE que dá suporte para os agentes cooperativos se adaptarem à ambientes dinâmicos. No SCARE, um meta agente gerencia a atribuição de agentes para atividades, de acordo com a sua análise da situação atual e o estado do agente e do ambiente. O meta agente é

responsável pelo processo de coordenação que é dividido em dois principais processos: a fase de tomada de decisão e a fase de negociação. Na fase de tomada de decisão o meta agente obtém e processa toda informação disponível, a fim de escolher a atividade mais apropriada a ser tomada pelos agentes, considerando a situação atual, a habilidade dos agentes e o estado do ambiente. Na fase de negociação, o meta agente procura pela melhor alocação de tarefas para os agentes através da observação das restrições de coordenação existentes. Este processo de coordenação pode ser executado por um único meta agente, como um processo centralizado, quando a comunicação e recursos computacionais tornam isto possível ou um meta agente pode ser replicado em diferentes agentes, gerenciando sub-redes de agentes conectados ao sistema multiagente.

Classificamos este trabalho como cooperação negociada, porque os agentes são cooperativos, mas eles têm que negociar a fim de executar suas tarefas, isto é feito através da fase de negociação. O trabalho usa a técnica de improvisação planejada, pois o meta agente considera a situação atual, a habilidade do agente e restrições de coordenação para adaptar a sua execução.

7.5.10 Rede de Contrato [HUH 99]

O protocolo de redes de contrato é um protocolo de interação para resolução de problemas cooperativos entre agentes. Ele é modelado com os mecanismos de contrato usados por empresas para governar a troca de bens e serviços. A rede de contrato fornece uma solução para o problema de conexão: encontrar um agente apropriado para trabalhar em uma determinada tarefa.

Um agente que quer a solução de uma tarefa é chamado de gerente; agentes que podem ser capazes de solucionar uma tarefa são chamados de potenciais contratantes. Do ponto de vista do gerente o processo ocorre da seguinte maneira:

- anuncia a tarefa que precisa ser executada;
- recebe e avalia a oferta de potenciais contratos;
- concede um contrato para um contratante apropriado;
- recebe e sintetiza resultados.

Sob o ponto de vista do contratante o processo ocorre da seguinte maneira:

- receber o anúncio da tarefa;
- avaliar sua capacidade para responder;
- responder (recusar, fazer uma oferta);
- executar a tarefa se sua oferta é aceita;
- reportar seus resultados.

Os papéis dos agentes não são especificados previamente. Qualquer agente pode agir como gerente fazendo anúncio de tarefas; qualquer agente pode agir como contratante, respondendo a um anúncio. Isto possibilita flexibilidade para decomposições futuras, pois um contratante para uma tarefa específica pode agir como gerente, solicitando ajuda de outros agentes para resolver partes da tarefa que lhe foi designada. As ligações gerentes-

contratantes forma uma hierarquia de controle para compartilhamento de tarefas e síntese de resultados.

Entendemos que a rede de contrato pode ser classificada como cooperação negociada porque os agentes têm o mesmo objetivo, e quando existe a necessidade de compartilhar recursos os acordos são mutuamente benéficos para ambos. Logo é cooperação, pois estão trabalhando para solução do mesmo objetivo, e existe negociação no momento que os agentes não têm habilidade para resolver a tarefa e têm que solicitar ajuda. A rede de contrato usa a técnica de planejamento, pois especifica antecipadamente como deve ocorrer a cooperação entre os agentes, não permitindo improvisação de comportamentos.

7.5.11 Sistemas de Quadro Negro [HUH 99]

Nos sistemas de quadro negro os agentes estão trabalhando cooperativamente, a fim de resolver um problema usando o quadro negro como um local de trabalho compartilhado para o desenvolvimento da solução. A solução do problema inicia quando o problema e dados iniciais são escritos no quadro negro. Os agentes observam o quadro negro, esperando por uma oportunidade de aplicar suas especialidades para o desenvolvimento da solução. Quando um agente encontra informação suficiente para fazer uma contribuição, ele coloca a contribuição no quadro negro. Este processo de adicionar contribuições ao quadro negro continua até o problema ter sido resolvido.

Classificamos o sistema de quadro negro como cooperação, porque os agentes estão trabalhando em conjunto, a fim de solucionar algum problema e eles podem resolver parte do problema utilizando seus próprios recursos. Ele se enquadra na técnica de planejamento, pois existem regras específicas de como interagir com o sistema de quadro negro.

7.5.12 Planejamento Distribuído [DUR 99]

Os mecanismos do planejamento distribuído já foram apresentados e discutidos na seção 7.2.1.1.2. Entendemos que todos os tipos de planejamento distribuídos são categorizados como cooperação, pois eles envolvem agentes que possuem o mesmo objetivo e constroem planos conjuntos. Cada agente é capaz de fazer sua parte no plano conjunto e não precisa competir por recursos. Como o próprio nome sugere, ele está classificado no modo de planejamento, porque existem regras prévias sobre como os agentes devem agir a fim de atingir seus objetivos.

A figura 7.5 mostra a classificação dos trabalhos apresentados no espaço de formas de coordenação proposto.

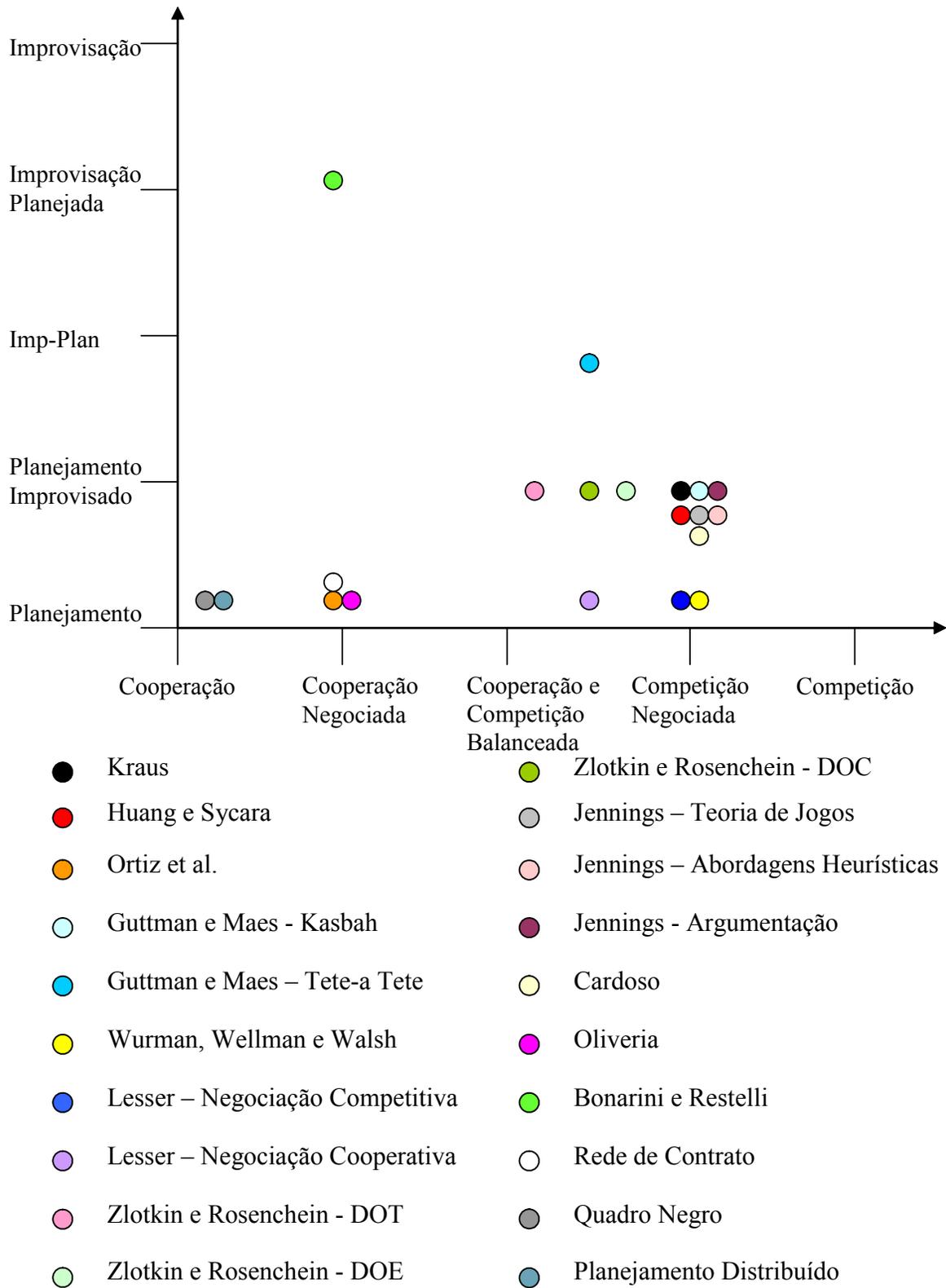


Figura 7.5: Espaço de Formas de Coordenação

7.5.13 Análise do Espaço de Formas de Coordenação

Separando os eixos dos estilos e técnicas em quatro domínios com seus estilos e técnicas predominantes teríamos: domínio competição e planejamento, domínio cooperação e planejamento, domínio cooperação e improvisação e domínio competição e improvisação (figura 7.6).

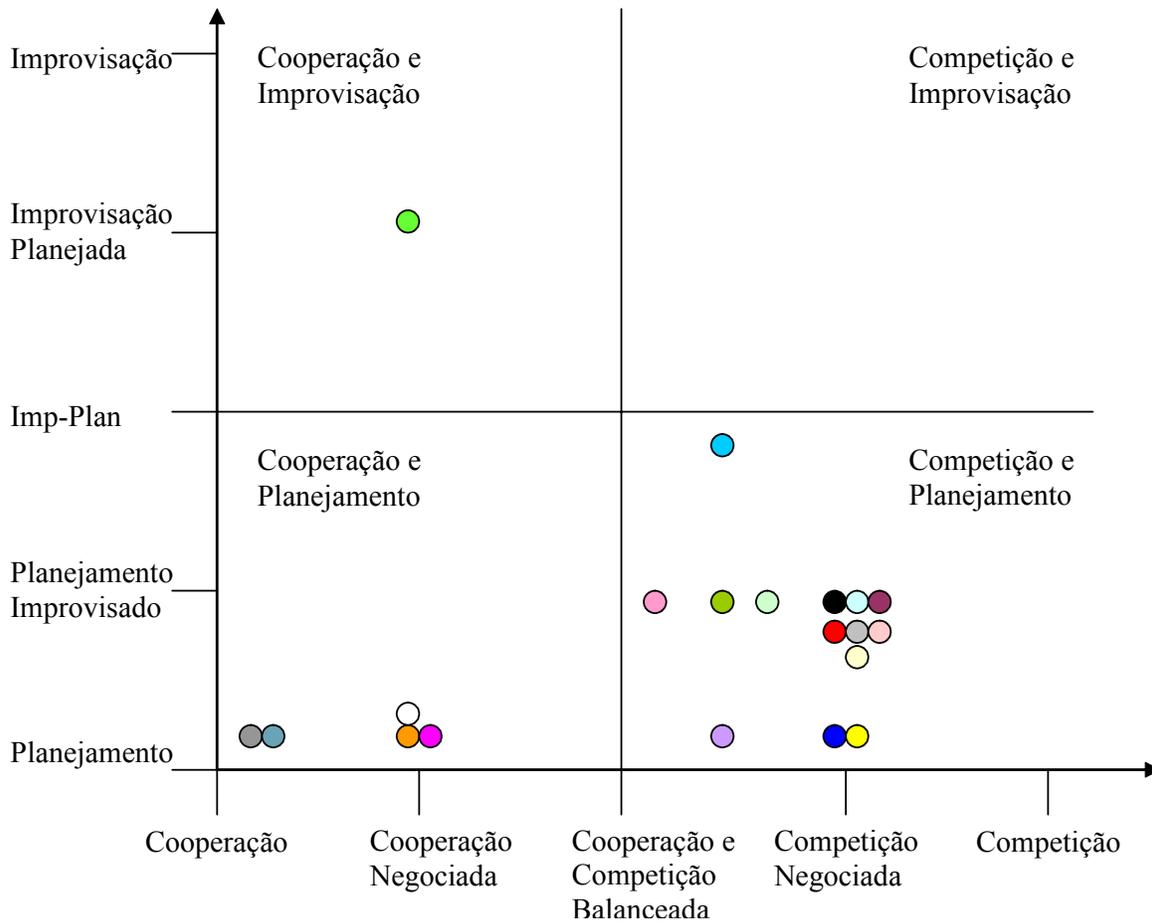


Figura 7.6: Espaço de Coordenação com Domínios Predominantes

A maioria dos trabalhos em coordenação se encontra no domínio da competição e planejamento. Foram pesquisados 20 trabalhos e os mesmos estão divididos nos domínios como mostra a tabela 7.1.

Tabela 7.1: Número de trabalhos em cada domínio

Domínio	Número de Trabalhos
Competição e Planejamento	14
Cooperação e Planejamento	5
Cooperação e Improvisação	1
Competição e Improvisação	0

A partir da tabela 7.1 podemos observar que as técnicas baseadas em regras pré-estabelecidas, que são representadas pelo nosso modo de planejamento, representam 95% das técnicas aplicadas à coordenação. Contudo, a maioria destes trabalhos se concentra no planejamento improvisado, sendo seguido pelo modo de planejamento tradicional.

Constatamos que a maioria dos trabalhos, mesmo sem querer, fornece ao agente algum mecanismo para improvisação, chamada por nós de implícita, pois é um tipo de improvisação que já foi previamente “planejada” ou em termos teatrais ensaiada e incorporada à representação formal.

Ainda analisando a tabela 7.1 podemos observar que o tipo de resultado predominante nos trabalhos pesquisados é competição, presente em 14 trabalhos. Destes, 9 trabalhos estão em competição negociada. Podemos constatar a partir da figura 7.6 que existe uma concentração de trabalhos entre a competição negociada e a cooperação e competição balanceada, por este motivo esta região foi nomeada de competição negociada cooperativa.

Observando a figura 7.6 podemos constatar que a maioria dos trabalhos se concentra na competição negociada. Como a competição negociada é equivalente à negociação competitiva [GUT 98a] [GUT 98b] (porque os agentes têm objetivos distintos e têm que negociar a fim de alcançar um acordo) este fato vem confirmar o que alguns autores têm salientado a respeito das pesquisas realizadas em coordenação. Estes autores dizem que as pesquisas em cooperação têm focado os aspectos de competição e não de cooperação entre os agentes [ZHA 2000] [ZHA 2001].

Nos trabalhos relacionados ao domínio da cooperação e planejamento, 3 estão na cooperação negociada e 2 estão na cooperação e todos eles se enquadram no modo de planejamento. O único trabalho que está no domínio da cooperação e improvisação, está na categoria de improvisação planejada.

Não foi encontrado dentro dos trabalhos pesquisados algum que se enquadrasse no resultado competição, tanto para a ênfase no modo de planejamento quanto para a ênfase no modo de improvisação. Isto ocorre porque a competição implica na inexistência de negociação entre as partes envolvidas, e todos os trabalhos pesquisados envolviam negociação de alguma maneira e, portanto não podem ser categorizados como competição pura.

Não encontramos nenhum trabalho que envolvesse negociação no domínio competição e improvisação. Em uma primeira vista isto parece estar de acordo com a idéia do que as pessoas entendem por negociação. Negociação, para a maioria das pessoas, é entendida como algo que deve ser planejado, evitando qualquer tipo de risco ou algo completamente inesperado. Contudo, apesar da improvisação ser “a arte do improvisado e do inesperado”, ela não implica que vai ser feito algo sem conexão com a situação atual e completamente sem nexos. A improvisação tem como ponto central o acontecimento do momento, ou seja, a situação atual, e algumas vezes ela pode ser incorporada a algo previamente ensaiado como é o caso da improvisação implícita. O fato da existência de níveis de improvisação é desconhecido pela maioria das pessoas.

Como mencionado anteriormente, somente um trabalho foi categorizado no domínio da cooperação e improvisação. A partir desta constatação pode surgir o seguinte questionamento: porque considerar improvisação como uma técnica adicional se somente

foi encontrado um trabalho envolvendo cooperação e improvisação? A primeira resposta para esta questão é que todos os trabalhos analisados são projetos específicos em coordenação de sistemas multiagentes, e os pesquisadores simplesmente não se preocupam com comportamento improvisado, eles querem que os agentes ajam corretamente e assumem que o planejamento é a melhor opção. Contudo, fazendo uma análise mais profunda destes trabalhos, encontramos algo bastante interessante: existe algum tipo de improvisação sendo utilizada, mas esta improvisação é parte de um método de planejamento. Isto pode ser comprovado na medida em que a maioria dos trabalhos utiliza planejamento improvisado para atingir seus estilos de coordenação. Uma segunda resposta é que não analisamos os trabalhos de coordenação aplicados a agentes de interface. Acreditamos que os trabalhos em agentes de interface irão ser classificados no domínio de cooperação e improvisação. Baseamos nossas crenças no estudo apresentado em Moraes [MOR 2001a] [MOR 2001b] que analisou vários projetos em agentes de interface e concluiu que improvisação está implícita e explicitamente representada em todos os trabalhos envolvidos com agentes animados ou personificados. Acreditamos que o mesmo irá acontecer para o caso da coordenação de agentes de interface.

Outra questão que pode surgir é, porque considerar competição sem negociação como um estilo se não existe nenhum trabalho classificado dessa forma? Acreditamos que muitos trabalhos envolvidos com agentes reativos e jogos podem ser classificados na categoria de competição sem negociação. Agentes reativos podem ser classificados como competitivos porque eles somente interagem com o ambiente e o ambiente impõe restrições às ações dos agentes. Estas restrições não podem ser negociadas, elas devem ser satisfeitas. A idéia básica de um jogo é que os jogadores estão competindo com o objetivo de ganhar alguma coisa. Não existe a possibilidade de negociação, todos pensam no seu próprio benefício e neste caso eles estão executando competição.

Com este espaço de coordenação apresentamos uma nova maneira de olhar o problema de coordenação, considerando-o como um espaço bi-dimensional onde estilos de coordenação e técnicas para atingir estes estilos são considerados. Também introduzimos uma nova idéia nos trabalhos em coordenação que é o uso da improvisação. Propomos examinar uma terceira dimensão para o nosso espaço de coordenação, a dimensão da presença do diretor. Nesta dimensão iremos trabalhar questões de coordenação centralizada e descentralizada e sua relação com a autonomia e heteronomia dos agentes.

7.6 Classificação dos Trabalhos relacionados no Espaço de Formas de Coordenação

Nesta seção iremos apresentar uma classificação para os trabalhos relacionados, descritos no capítulo três, dentro do espaço de formas de coordenação proposto.

7.6.1 Teatro virtual

Não existe interação entre os agentes, ou seja, os agentes não interagem entre si, mas sim com um usuário. Desta maneira, não é necessária a coordenação. Contudo, podemos dizer que a maneira como eles produzem seus comportamentos é através de uma improvisação planejada, pois eles implementam a idéia de improvisação, mas escolhem os

comportamentos a serem improvisados através de um plano de controle e de um escalonador.

7.6.2 Empresa Extempo Systems Inc.

Devido à influência que recebeu dos trabalhos desenvolvidos no Teatro Improvisacional, os *imp characters* também não envolvem aspectos de coordenação, porque não ocorre interação entre dois, ou mais, agentes artificiais. Os *imp characters* interagem diretamente com o usuário e o seu método de escolha de comportamento também poderia ser categorizado como improvisação planejada.

7.6.3 Projeto Oz

No projeto Oz identificamos a presença da coordenação através do gerenciador do drama. Os agentes que fazem parte do drama são criados e excluídos pelo gerenciador do drama, não existe uma interação de negociação ou cooperação entre os agentes criados. Podemos entender que existe uma relação de cooperação entre o gerenciador do drama e o agente por ele criado para realizar uma determinada tarefa. O modo como o gerenciador do drama conduz uma experiência dramática é através de uma busca adversária, que se assemelha a uma busca em árvore, com uma função de avaliação, ou seja, o método utilizado pelo gerenciador do drama é planejado, mas a função de avaliação considera critérios dramáticos que se modificam com o tempo, o que poderia ser categorizado como uma tentativa de se adaptar a situação atual, ou seja improvisar. Desta maneira, o modo mais adequado seria planejamento improvisado.

7.6.4 Projeto Improv

No projeto Improv um agente pode interagir com outros agentes fazendo uso de suas regras de comportamento que são previamente descritas por um autor. Os agentes se comunicam através de um quadro negro compartilhado. As ações de coordenação dos agentes são dadas por suas regras de comportamento e têm como objetivo produzir uma cooperação entre os agentes, que procuram interagir da melhor maneira possível a fim de atingir seu objetivo que é a própria interação. Eles alcançam esta cooperação através do quadro negro que é um modo pertencente à categoria do planejamento, visto que possui suas regras de acesso bem definidas.

7.7 Espaço de Formas de Coordenação Estendido

Como estamos interessados em entender coordenação como direção improvisacional é interessante estender o espaço de formas de coordenação anteriormente proposto para que o mesmo abranja o aspecto de distribuição de coordenação. Neste sentido, o espaço de formas de coordenação deixa de ser bidimensional para se tornar tridimensional como mostra a figura 7.7.

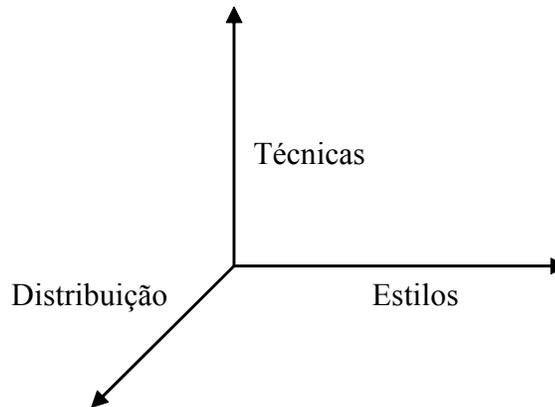


Figura 7.7: Eixos do Espaço de Formas de Coordenação Estendido

Na próxima seção iremos discutir o que deve ser considerado no eixo da distribuição.

7.7.1 Eixo da Distribuição da Coordenação

Vários pesquisadores têm se preocupado com as questões de centralização e descentralização para coordenação de agentes, dentre eles podemos destacar Durfee [DUR 99] e Zhang et al. [ZHA 2000] [ZHA 2001]. De acordo com estes autores, podemos ver centralização como a presença de um processo ou agente capaz de realizar coordenação. Podemos dizer que um processo centralizado envolveria a existência de um agente específico, um agente diretor, responsável pela coordenação. Por outro lado, a descentralização pode ser entendida como se vários agentes estivessem envolvidos no processo de coordenação. Neste caso, não existiria um agente diretor, mas vários diretores. Nesta visão cada agente seria o seu próprio diretor e decidiria como interagir e se coordenar com os demais agentes.

Entendemos que estes dois aspectos estão diretamente relacionados com a “*nomia*” dos agentes, ou seja, com as regras dos agentes. Em um extremo temos a autonomia (significando regras próprias) e em outro extremo a heteronomia (significando as regras dos outros). Neste sentido, um agente completamente autônomo teria a capacidade de se dirigir e coordenar suas ações e interações com outros agentes. Já um agente completamente heterônimo seria completamente dependente das regras e direções fornecidas por outros agentes. Assim, poderíamos definir graus de direção que variam desde a inexistência de um diretor central, mas de vários diretores descentralizados, distribuídos, até a existência de um diretor central, como mostra a figura 7.8.

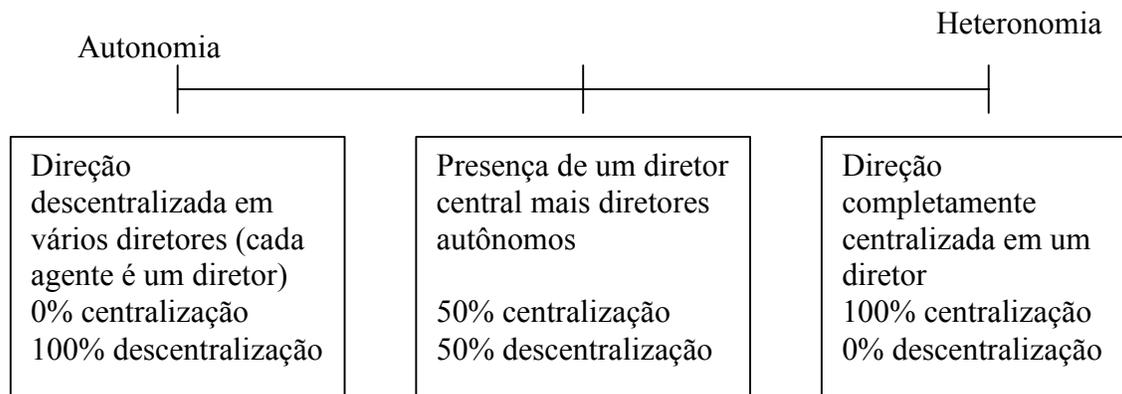


Figura 7.8: Visão inicial do eixo da distribuição

O eixo apresentado na figura 7.8 é uma visão inicial sobre as componentes que fazem parte do eixo da direção.

7.8 Considerações sobre o Capítulo

Neste capítulo apresentamos os conceitos envolvidos em coordenação e desenvolvemos uma proposta para classificação de trabalhos de coordenação chamada de espaço de formas de coordenação. O espaço proposto foi inicialmente composto por dois eixos: o dos estilos (abrangendo os conceitos de cooperação e competição) e o das técnicas (abrangendo os conceitos de planejamento e improvisação). Foram criados critérios para as categorias destes eixos e realizada uma classificação de vinte trabalhos desenvolvidos na área. Estes trabalhos foram escolhidos por seus autores serem pessoas com experiência em projetos envolvendo coordenação de agentes. Contudo, como mencionado anteriormente, alguns trabalhos em coordenação de agentes reativos e jogos não foram considerados o que apresentou a baixa ou inexistência de trabalhos em algumas áreas do espaço de formas de coordenação.

Como a coordenação como direção improvisacional é uma característica importante a ser analisada, propusemos a inserção de um novo eixo ao espaço de formas criando, assim, o espaço de formas estendido. Entendemos que o eixo da distribuição está diretamente relacionado à autonomia e heteronomia dos agentes, o que resulta em dois aspectos de coordenação que serão abordados no próximo capítulo, a coordenação centralizada (onde existe um diretor improvisacional envolvido) e descentralizada (onde não existe a figura específica do diretor, mas sim de atores que agem também com seus próprios diretores).

8 AGENTES DELIBERATIVOS BASEADOS EM ARQUITETURA BDI

Segundo Wooldridge [WOO 2000], agentes são entidades que agem sobre o ambiente no qual habitam. Esta atuação tem como objetivo a execução de ações que modificam o ambiente habitado pelo agente, e estas mudanças levam o agente a atingir seu objetivo atual. Um agente é dito racional se ele escolhe as melhores ações para serem executadas em um determinado momento, levando em consideração suas crenças atuais sobre o mundo.

Para tomarem decisões, os agentes racionais precisam deliberar sobre o problema. Desta maneira, a deliberação tem um papel importante no projeto de agentes racionais inseridos no mundo real [RAO 91a] [RAO 91b]. A deliberação leva a formação de intenções, isto é, planos de ações que o agente está comprometido a alcançar. É através da execução destes planos de ações que o agente consegue atingir seus objetivos.

Analisando as definições apresentadas acima, podemos observar a existência de três aspectos importantes para agentes racionais que realizam deliberação: crenças, utilizadas para escolher as melhores ações a serem executadas; desejos, utilizados para guiar a execução das ações pelo agente; e intenções (ou objetivos) que devem ser alcançados pelo agente. Estes três aspectos são de fato os aspectos considerados no modelo de agência racional BDI (*Belief-Desire-Intention*), modelo de crenças-desejos-intenções.

O modelo BDI foi originalmente desenvolvido por Bratman [BRA 99a] [BRA 88] e se baseia nas idéias de sua teoria de raciocínio prático. De acordo com Wooldridge [WOO 99], o raciocínio prático envolve dois processos importantes: o de decidir quais objetivos se quer atingir e o de como se quer atingir estes objetivos. O primeiro processo, como visto acima, é conhecido como deliberação e o segundo processo como raciocínio meio-fim. O processo de decisão do raciocínio prático inicia se tentando entender quais opções estão disponíveis para o agente. Após ter um conjunto de alternativas (os desejos), o agente deve escolher entre elas e se comprometer com alguma. As opções escolhidas se transformam em intenções, que irão determinar as ações dos agentes. O processo se repete, fazendo com que as intenções alimentem o futuro raciocínio prático do agente. O processo pára quando o agente atinge seu objetivo.

A idéia do raciocínio prático foi introduzida por Bratman [BRA 88] com a finalidade de tratar o problema da limitação de recursos para agentes que habitam mundos complexos. Neste sentido, o papel das intenções é restringir a quantidade de raciocínio prático futuro que o agente deve realizar. As intenções são vistas como estruturadas dentro

de planos maiores e levam o agente a ter um determinado comprometimento com a ação futura.

Segundo Wooldridge [WOO 2000], o que torna o modelo BDI particularmente interessante é que ele combina três elementos importantes: uma teoria a respeito da ação racional, a aplicação com sucesso do modelo em vários domínios complexos através da especificação de arquiteturas e a formalização de lógicas de BDI.

Nas próximas seções iremos abordar cada um destes elementos, ressaltando os aspectos importantes da teoria de ação racional como intenção, planos, comprometimentos, crenças e desejos. Iremos mostrar algumas arquiteturas BDI desenvolvidas com base na teoria de ação racional e apresentaremos alguns trabalhos que enfocam a especificação da lógica BDI.

8.1 Aspectos Relacionados à Teoria de Raciocínio Prático

Bratman [BRA 99a] baseia sua teoria de raciocínio prático na racionalidade humana e entende que o aspecto importante da sua teoria está em considerar o fenômeno de planos e planejamento e a sua relação com as intenções. Para Bratman os seres humanos são agentes de planejamento que têm capacidade de construir planos de ações mais ou menos complexos para programar suas atividades. Esta capacidade mostra uma característica do processo de deliberação dos agentes de planejamento que é a de ajudar a escolher não somente as ações presentes, mas também as ações futuras. Os planos também permitem coordenar as atividades individuais e coletivas dos seres humanos. Contudo, Bratman salienta que os planos devem ser parciais, pois é impossível prever todos os acontecimentos do futuro. Esta característica é fundamental, pois permite completar o plano na medida em que ocorrem os cursos de ação.

As intenções dirigem o comportamento futuro dos agentes e estão relacionadas a um comprometimento com ações futuras. O comprometimento relacionado às intenções possui duas dimensões. A primeira dimensão se preocupa com a relação entre intenção e ação e é chamada de dimensão de vontade. A dimensão de vontade indica que as intenções são controladores de conduta da ação. Tanto as intenções quanto os desejos são pró-attitudes motivacionais que, junto com as crenças podem levar à ação. Contudo, enquanto a intenção é um controlador de conduta da ação, o desejo é simplesmente um potencial influenciador da ação [BRA 99a] [BRA 99b] [WOO 2000]. Desejos possuem uma conexão mais fraca com a racionalidade do agente, pois os desejos podem ser mutuamente inconsistentes e inconsistentes com as crenças, enquanto as intenções não [SIN 94].

A segunda dimensão do comprometimento são os papéis que as intenções dirigidas ao futuro têm no período entre a sua formação inicial e sua eventual execução. Estes papéis constituem o comprometimento centrado no raciocínio. Existem dois principais fatores. O primeiro é que a intenção resiste à reconsideração, ela tem uma característica de estabilidade ou inércia. O segundo fator está relacionado ao papel que a intenção tem entre o raciocínio realizado agora e o raciocínio futuro. O agente freqüentemente irá raciocinar das intenções que ele já possui para as intenções futuras. As intenções devem ser consistentes umas com as outras e com as crenças do agente [BRA 99a] [SIN 94].

Intenções relacionadas às ações futuras são elementos em planos maiores, planos que facilitam coordenação tanto social quanto própria, planos que ajudam a fazer com que

as deliberações anteriores dirijam a conduta posterior. Planos são vistos por Bratman como estados mentais, envolvendo algum tipo de comprometimento, e não como estruturas abstratas. Neste sentido, planos são intenções amplas e compartilham as mesmas propriedades das intenções: resistem à reconsideração, são controladores de conduta e fornecem entradas importantes para raciocínio e planejamento futuro. Além destas propriedades, os planos possuem duas características importantes [BRA 88] [BRA 99a]:

- os planos são parciais: a medida em que o tempo passa, o plano é preenchido com especificações de meios, passos preliminares e cursos de ação mais específicos;
- os planos normalmente têm uma estrutura hierárquica: planos relacionados a fins, incorporam planos relacionados a meios e passos preliminares. Deste modo, pode-se deliberar sobre partes do plano enquanto mantém outras partes inalteradas.

A fim de suportar a coordenação e estender a influência na conduta futura, os planos devem satisfazer as seguintes demandas [BRA 88] [BRA 99a]:

- restrições de consistência: para coordenar as atividades sobre o tempo, o plano deve ser internamente consistente. O plano deve servir para o mundo em que o agente se encontra, desta maneira, o plano deve ser consistente com as crenças do agente;
- coerência meio-fim: embora os planos sejam parciais, eles ainda devem ser preenchidos de maneira apropriada à medida que o tempo passa. O plano deve ser preenchido com subplanos contendo os meios, passos preliminares, e cursos de ação relativamente específicos, planos que sejam pelo menos tão extensíveis quanto se acredita ser necessário para se realizar o plano.

As propriedades das intenções e dos planos ajudam a focar a deliberação na medida em que estabelecem padrões de relevância para as opções que devem ser consideradas na deliberação. Desta maneira, as intenções e os planos restringem as possíveis soluções para o problema, pois fornecem um filtro que indica quais opções são admissíveis. Para escolher entre as opções admissíveis, o agente faz uso das suas crenças sobre o mundo e dos seus desejos atuais. As crenças do agente indicam o que o agente toma como sendo verdadeiro para o seu mundo em um determinado momento. Os desejos são pró-attitudes que motivam o agente a realizar uma determinada ação.

Considerando estas idéias, Bratman [BRA 99a] salienta que o raciocínio prático é composto por dois níveis. Em um primeiro momento, as intenções e os planos colocam problemas e fornecem um filtro nas opções que podem solucionar estes problemas. Em um segundo momento, as crenças e os desejos são levados em consideração para auxiliar na escolha entre as opções relevantes e admissíveis. Esta estrutura é essencial para que as intenções e os planos executem o seu papel como facilitadores de coordenação das ações dos agentes.

Podemos resumir os conceitos apresentados nesta seção através do entendimento de Rao e Georgeff [RAO 91a] sobre o raciocínio prático. Segundo Rao e Georgeff, na visão de Bratman, intenções são planos de ações que o agente está comprometido a alcançar. Este comprometimento é importante para estabelecer o escopo do processo de deliberação. Em primeiro lugar as intenções produzem os problemas de decisão que o agente necessita considerar. Em segundo lugar, as intenções restringem o processo de deliberação, porque desconsideram opções que são conflitantes com as intenções existentes. Enquanto as

intenções ajudam a determinar e restringir as opções abertas para deliberação, o processo de deliberação gera novas intenções. Através do raciocínio meio-fim, uma intenção caminha no sentido de um resultado final enumerando todos os meios alternativos de alcançar este fim. O agente deliberando sobre estas alternativas escolhe o melhor caminho, se comprometendo com ele através da criação de uma nova intenção. Desta maneira, o processo de deliberação é uma atividade limitada a recursos contínua ao invés de uma teoria de decisão exaustiva.

8.2 Aplicações da Teoria de Raciocínio Prático: Arquiteturas BDI

Desde meados de 1980 a teoria de raciocínio prático, ou modelo BDI, tem sido amplamente estudada e implementada através de diversas arquiteturas. A primeira implementação foi IRMA (*Intelligent Resource-bounded Machine Architecture*). IRMA foi desenvolvida por Bratman, Israel e Pollack [BRA 88] com o objetivo de proporcionar uma implementação para a teoria de raciocínio prático de Bratman. Contudo, a implementação mais conhecida é o *Practical Reasoning System* (PRS) [GEO 89] e os seus descendentes como o dMars [D'IN 97]. No PRS, um agente tem uma estrutura de dados que explicitamente corresponde a crenças, desejos e intenções. As crenças de um agente PRS são diretamente representadas na forma de fatos Prolog e os desejos e as intenções no PRS são executadas através do uso de uma biblioteca de planos. Uma biblioteca de planos, é uma coleção de planos. Cada plano é uma receita que pode ser usada por um agente para alcançar algum estado específico. Um plano PRS é caracterizado por um corpo e uma condição de ativação. O corpo do plano é um curso de ação que pode ser usado para atingir algum estado e a condição de ativação define as circunstâncias nas quais um agente deve considerar o plano. O controle no PRS é realizado pela contínua atualização dos estados internos do agente e observação se o plano possui condições de ativação correspondentes a estas crenças. O conjunto de planos ativos corresponde aos desejos dos agentes. Cada desejo define um curso de ação possível que um agente pode seguir. Em cada ciclo de controle, o PRS pega um destes desejos, e o coloca na pilha de execução. A pilha de execução corresponde aos desejos escolhidos pelos agentes, ou seja, corresponde às intenções do agente.

Nesta seção iremos apresentar abordagens clássicas do modelo BDI e algumas extensões ao modelo como o tratamento de intenções conjuntas e inclusão de restrições. Iremos apresentar como abordagens clássicas a arquitetura IRMA e um descendente do PRS, a arquitetura dMARS. Também iremos apresentar como abordagem clássica um esquema de uma arquitetura genérica para agentes BDI proposta por Wooldridge [WOO 2000].

8.2.1 Arquitetura IRMA

O ponto de partida para a construção da arquitetura IRMA [POL 92] foi a idéia proposta por Bratman de que os agentes se comprometem a planos, que enquadram e restringem o seu raciocínio futuro. Os planos indicam sobre o que o agente deve raciocinar. O agente deve então descobrir como executar os planos para os quais ele está comprometido, sem considerar completamente as opções que são incompatíveis com os

seus compromentimentos existentes. Desta maneira, os agentes devem executar raciocínio meio-fim e escolher entre cursos de ação alternativos. Uma arquitetura para agentes racionais deve incorporar estas duas capacidades e também deve mostrar como estas capacidades interagem [BRA 88]. A arquitetura apresentada em [BRA 88] faz isto (figura 8.1).

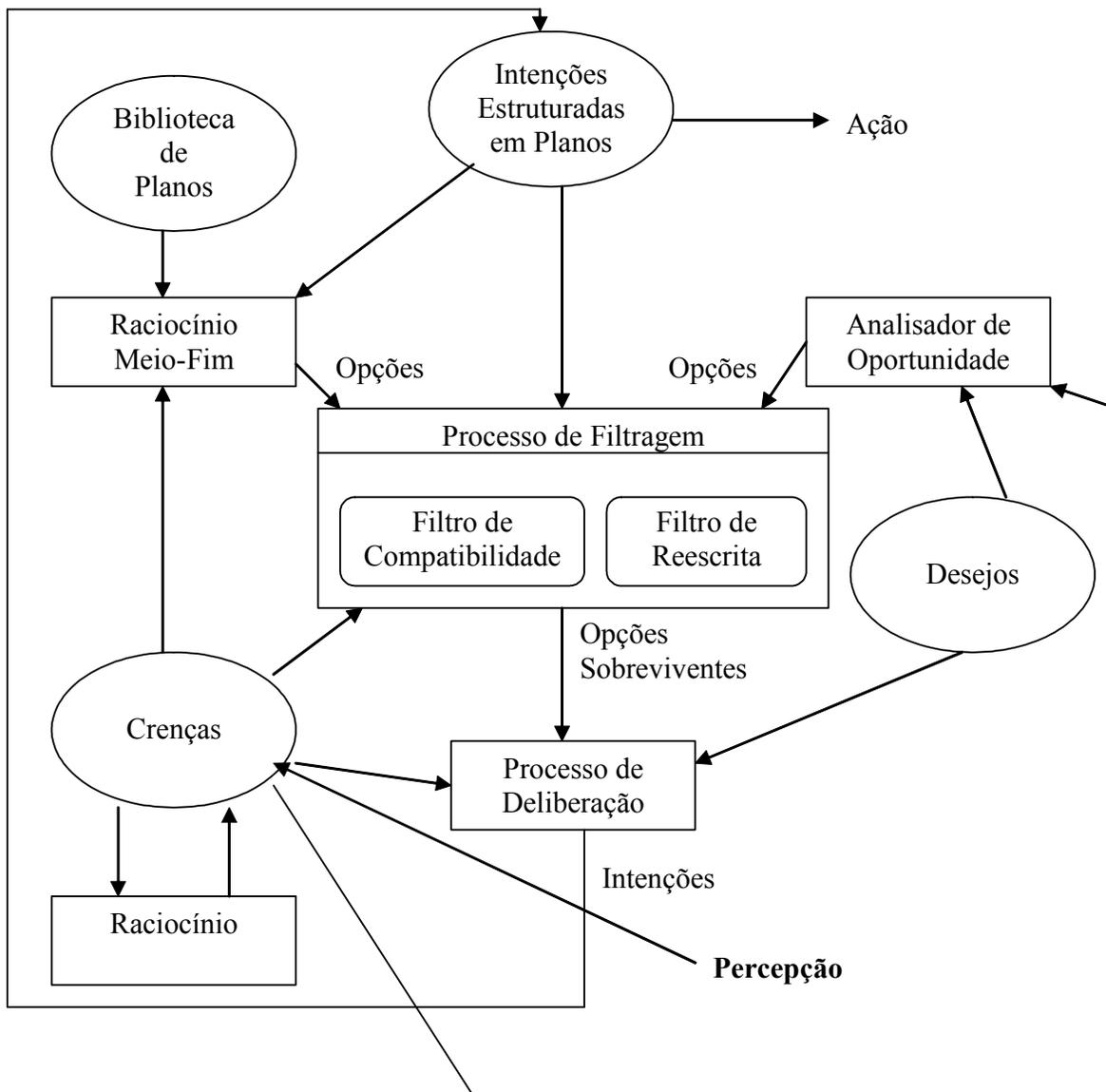


Figura 8.1: Uma arquitetura para agentes limitados a recursos [BRA 88]

Uma arquitetura para raciocínio prático em agentes com recursos limitados pode ser classificada como uma arquitetura de crenças, desejos, intenções (BDI), pois inclui representações diretas das crenças, desejos e intenções dos agentes. As intenções dos agentes são estruturadas dentro de planos maiores. Existe uma distinção entre os planos que o agente realmente adotou, chamado de Intenções Estruturadas dentro de Planos, e os

planos como receitas, ou operadores, que o agente tem conhecimento, chamado de Biblioteca de Planos. A biblioteca de planos pode ser vista como um subconjunto das crenças dos agentes, ou seja, crenças sobre quais ações podem ser úteis para se alcançar os objetivos, dadas determinadas condições. Os planos chamados de Intenções Estruturadas dentro de Planos são uma coleção estruturada de crenças e intenções [POL 92] [POL 90].

A arquitetura IRMA se baseia na arquitetura de agentes com recursos limitados e inclui todos os seus componentes. Contudo, a arquitetura não representa de forma explícita as crenças, desejos e biblioteca de planos, e o analisador de oportunidades está embutido no mecanismo de reescrita. Através da figura 8.2 podemos visualizar a arquitetura IRMA.

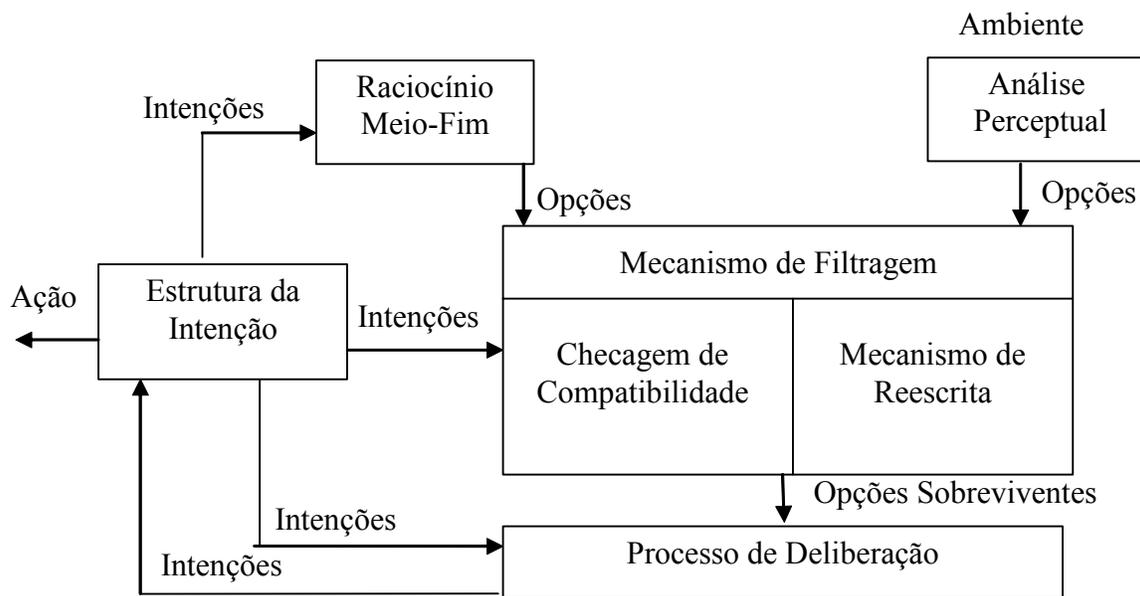


Figura 8.2: Arquitetura IRMA [POL 92]

Os componentes da arquitetura IRMA são:

- raciocínio meio-fim: possibilita ao agente raciocinar sobre como atingir seus objetivos;
- processo de deliberação: possibilita ao agente decidir quais objetivos devem ser adotados. O processo de deliberação é alguma rotina que recebe um conjunto de opções (chamadas na figura de opções sobreviventes), pesa as suas saídas e seleciona a opção ou opções que excedem algum limite. As opções que são selecionadas pelo processo de deliberação se tornam novas intenções dos agentes;
- mecanismos de filtragem: verifica se uma opção é compatível com os planos existentes dos agentes. O mecanismo de filtragem é composto por dois módulos. O módulo de checagem de compatibilidade e o mecanismo de reescrita. A checagem de compatibilidade checa as opções para determinar a compatibilidade com os planos existentes. As opções compatíveis são as opções sobreviventes, que são passadas ao processo de deliberação. O mecanismo de reescrita codifica a sensibilidade do agente para problemas e oportunidades no seu ambiente. Caso uma

opção seja incompatível, ou seja, não sobreviva a checagem de compatibilidade, ela pode ser reconsiderada. Ela será reconsiderada se disparar uma sensibilidade do agente. A sensibilidade do agente é codificada através de classes de características ambientais às quais o agente é sensível. O mecanismo de reescrita deve ser cuidadosamente projetado para incorporar o grau certo de sensibilidade a problemas e oportunidades que surgem no seu ambiente.

As opções que são consideradas no mecanismo de filtragem podem ser sugeridas por mudanças no ambiente ou podem ser sugeridas pelo raciocínio meio-fim como maneiras possíveis de se alcançar os objetivos adotados. O desafio do mecanismo de filtragem é balancear a sensibilidade a mudanças no ambiente contra a estabilidade de raciocínio dos planos adotados pelo sistema.

De acordo com Pollack [POL 99], a idéia chave de IRMA é que os agentes não devem realizar processos de deliberação completos para novas opções que estão em conflito com os seus comprometerimentos existentes, a menos que estas opções possam ser facilmente reconhecidas como potencialmente especiais de alguma maneira. Para alcançar este balanço entre comprometerimentos a planos existentes e sensibilidade a novas opções particularmente importantes, foi colocado na arquitetura IRMA um mecanismo de filtragem composto por dois módulos, um responsável pela checagem de compatibilidade e outro composto por um mecanismo de reescrita.

Os agentes IRMA podem algumas vezes se comportar de uma maneira subótima, ou seja, podem executar ações diferentes daquelas que seriam selecionadas se o agente tivesse desenvolvido um processo de deliberação completo. A arquitetura pode executar raciocínio que não deveria, ou seja, ela pode executar raciocínio sobre uma opção e depois decidir em manter a opção atual. Contudo, um agente IRMA pode ter um nível de performance geral melhor do que um agente que sempre executa um processo de deliberação completo para cada nova opção que surge, pois o tempo gasto em tais deliberações normalmente ocasiona a perda de prazos e oportunidades.

Desta maneira, os agentes IRMA são agentes de planejamento [POL 92]. O comprometimento dos agentes aos seus planos tem um papel amplo na redução dos problemas de decisão. Este comprometimento é garantido na arquitetura pela checagem de compatibilidade. Contudo, os agentes de planejamento também devem estar atentos às oportunidades que surgem e isto é implementado através do mecanismo de reescrita que descreve quais sensibilidades o agente pode ter em relação aos acontecimentos do ambiente.

Caso o agente queira se coordenar e se comunicar com outros agentes é extremamente útil que os agentes sejam planejadores. Existem duas razões para isto:

- a coordenação entre os agentes parece possível, pois eles podem contar que os agentes irão agir de maneira estável (comprometimento a sua intenção);
- a comunicação entre os agentes é facilitada na medida em que o agente pode raciocinar sobre os planos de outros agentes.

8.2.2 Arquitetura dMARS

A arquitetura dMARS (Sistema de Raciocínio Multiagente) é uma implementação em C++ do sistema de raciocínio prático (PRS) que tem como objetivo fornecer um modelo formal de um sistema que define precisamente as estruturas de dados chave presentes dentro da arquitetura e as operações que manipulam estas estruturas. Assim como a arquitetura PRS, dMARS tem suas raízes conceituais no modelo de raciocínio prático de crenças-desejos-intenções desenvolvido por Bratman [BRA 99a] [BRA 88]. Desta maneira, a arquitetura de dMARS se baseia na arquitetura PRS e pode ser visualizada na figura 8.3.

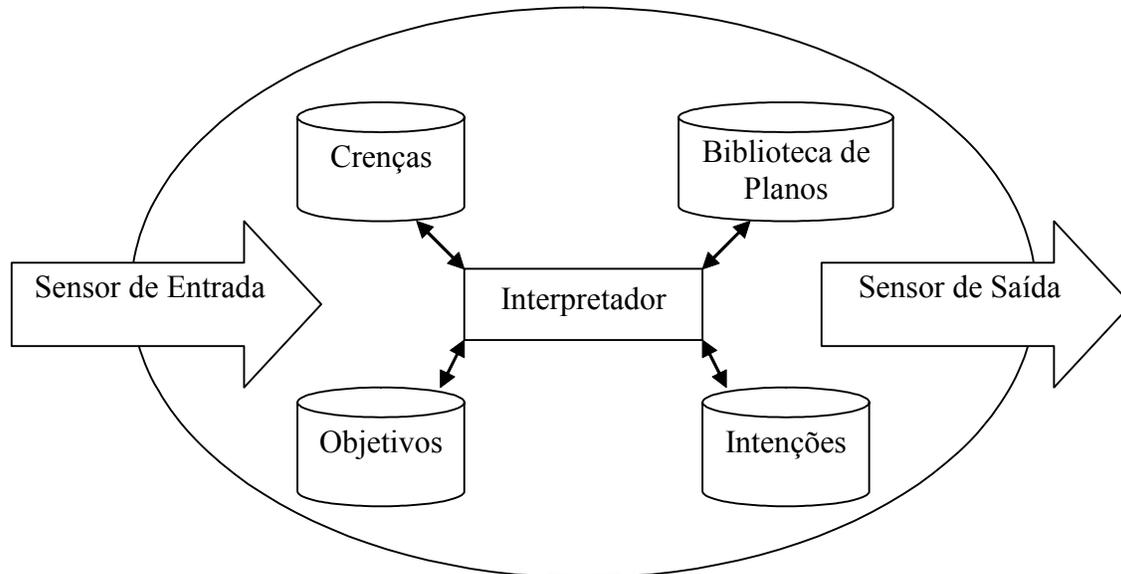


Figura 8.3: Visão Geral da Arquitetura dMARS [D'IN 97]

As crenças de um agente correspondem à informação que o agente tem sobre o mundo, que pode ser incompleta ou incorreta. Crenças podem ser tão simples quanto variáveis, mas os agentes BDI normalmente representam crenças simbolicamente, como fatos Prolog. Os desejos de um agente (ou objetivos do sistema) correspondem intuitivamente às tarefas alocadas para ele. Os agentes devem se fixar em um subconjunto dos desejos e comprometer recursos para realizar estes desejos. Os desejos escolhidos são as intenções do agente. O agente interrompe a busca pela sua intenção no momento em que ele acredita que a intenção já tenha sido satisfeita ou que a intenção não pode mais ser alcançada.

O modelo BDI é operacionalizado em agentes dMARS através de planos. Cada agente tem uma biblioteca de planos que é um conjunto de planos, ou receitas, especificando cursos de ação que podem ser tomadas pelo agente a fim de alcançar suas intenções. A biblioteca de planos de um agente representa seu conhecimento procedural, ou seja, conhecimento sobre como alcançar os estados desejados.

Os agentes dMARS monitoram o mundo e seu próprio estado interno e qualquer evento percebido é passado para a fila de eventos. O interpretador é responsável por gerenciar todas as operações dos agentes. Desta maneira, ele executa continuamente o seguinte ciclo [D'IN 97]:

- observa o mundo e o estado interno do agente, e atualiza a fila de eventos para refletir os eventos que foram observados;
- gera novos desejos (tarefas) possíveis, encontrando planos cuja condição de ativação do evento combine com um evento da lista de eventos;
- seleciona do conjunto de planos que combinam um plano para execução (um meio pretendido);
- coloca o meio pretendido em uma pilha de intenções existentes ou em uma nova pilha de intenções, considerando se o evento é um subobjetivo;
- seleciona uma pilha de intenção, pega o plano (meio pretendido) que está no topo, e executa o próximo passo do seu plano corrente, coloca este subobjetivo na fila de eventos.

Desta maneira, quando um plano inicia sua execução, seus subobjetivos serão colocados na fila de eventos que irão fazer com que os planos que alcancem este subobjetivo se tornem ativos e assim por diante. Este é o modelo básico de execução de agentes dMARS. Os agentes não fazem nenhum tipo de pré-planejamento, pois todos os planos devem ser gerados pelo programador do agente em tempo de execução. O planejamento realizado pelos agentes consiste da expansão de subobjetivos sensíveis ao contexto, que é deferido até o ponto do tempo no qual o subobjetivo é selecionado para execução.

As crenças em dMARS são como fatos Prolog, ou seja, são fórmulas atômicas positivas e negativas não contendo nenhuma variável. As ações que os agentes podem executar são classificadas como externas (no caso do domínio de ação ser o ambiente fora do agente) ou interna (no caso do domínio de ação ser o próprio agente). Ações externas são especificadas como se fossem chamadas a procedimentos ou invocação de métodos. Ações internas podem ser de dois tipos: adicionar ou remover uma crença da base de dados.

Os planos consistem de cinco componentes: uma condição de ativação (ou evento de ativação); um contexto ótimo (uma situação) que define as pré-condições de um plano, ou seja, o que o agente deve acreditar para que o plano seja executado; uma condição de manutenção que deve ser verdadeira para o plano continuar executando; um conjunto de ações internas que são executadas se o plano tem sucesso; e um conjunto de ações internas que são executadas quando o plano falha. O corpo do plano é representado através de uma árvore. Os nodos representam estados e os arcos representam um objetivo, ou uma ação interna, ou uma ação externa. Dizer que um plano foi executado com sucesso implica em percorrer a árvore da raiz até qualquer nodo folha [D'IN 97].

Uma condição de ativação pode ser disparada por quatro tipos de eventos: a aquisição de uma nova crença; a remoção de uma crença; o recebimento de uma mensagem ou a aquisição de um novo objetivo. A combinação de uma condição de ativação com um evento presente na fila de eventos e a combinação de um contexto com as crenças atuais do agente gera um conjunto de planos candidatos. O agente então seleciona um plano e cria uma instância do mesmo. Uma instância do plano contém uma cópia do plano original e outros elementos. Estes elementos são:

- o ambiente do plano (quaisquer instanciações que foram geradas no curso de execução do plano);

- o estado atual alcançado no plano (inicialmente a raiz do corpo do plano);
- o conjunto de ramos que ele pode tentar percorrer a partir do seu estado;
- o ramo (caminho) que ele está percorrendo;
- um identificador único que identifica a instância do plano para o agente proprietário do conjunto de todos os identificadores ativos;
- o estado do plano (ativo, indicando que o plano é parte de uma intenção ou inativo indicando que o plano está temporariamente suspenso).

Quando um ramo (caminho) não pode ser percorrido (porque uma ação ou subobjetivo falha), o ramo falha e é removido do conjunto de ramos possíveis. Se o ramo que o agente está tentando percorrer é definido, o agente já escolheu que ramo executar a seguir, mas, se o ramo está indefinido significa que nenhuma escolha foi feita.

Uma intenção no dMARS é uma seqüência de instâncias de planos. Uma intenção é criada em resposta a um evento externo. A intenção criada contém a instância gerada do plano. Se este plano cria um evento interno, ao qual o agente responde com um outro plano, o novo plano é adicionado à intenção. Desta maneira, o plano que está no topo da pilha de intenção é o plano que irá ser executado primeiro na intenção [D'IN 97].

Desta maneira, planos no dMARS são seqüências de ações e objetivos com pontos de escolha. Isto indica que, a qualquer momento, podem existir mais de um caminho a percorrer a fim de completar o plano. Intenções são planos que estão sendo executados atualmente. As intenções determinam que ações os agentes tomam e podem levar a geração de novos subobjetivos, que pode ocorrer durante o percurso do agente executar o seu plano.

O modelo formal para dMARS especifica como os planos são determinados inicialmente como relevantes e aplicáveis, como um plano é escolhido e como ele é usado. Para fazer isto, um agente dMARS consiste de uma biblioteca de planos, uma função de seleção de intenção, uma função de seleção de eventos e uma função de seleção de planos. Ele também possui uma função de seleção de substituição, para escolher entre instanciações alternativas possíveis, e uma função para selecionar que caminho no plano deve ser realizado a seguir.

Podem existir dois casos quando um plano tem sucesso. Se existem mais planos na intenção, a substituição atual do ambiente é atualizada para incluir as instanciações apropriadas do plano alcançado e incluir o ambiente do novo plano na pilha. A instância do plano que teve sucesso é então removida da pilha e o próximo plano a ser executado (o segundo do topo da pilha) é reativado. Se não existem mais planos, a intenção tem sucesso e pode ser removida assim como o evento externo que o gerou.

Se um caminho falha, mas existem outros caminhos possíveis, eles devem ser executados. Se não existem mais alternativas para o plano então o plano falha. Quando o plano é o único da pilha, a intenção falha completamente. Quando um plano falha, deve-se atualizar o registro para que a instância do plano que falhou não seja mais executada.

Através destas explicações podemos observar que as estruturas de controle do dMARS são explicitamente codificadas nos planos. Os planos no dMARS são vistos como programas a serem executados através de suas ações primitivas. Ao contrário da arquitetura IRMA, onde as intenções são estruturas dentro de planos maiores, as intenções no dMARS

são compostas por instâncias de planos. Ambas arquiteturas utilizam as idéias do raciocínio prático de Bratman, mas tratam alguns aspectos de forma diferenciada.

8.2.3 Esquema de Wooldridge para uma Arquitetura BDI Genérica

Wooldridge [WOO 99] apresenta um esquema de uma arquitetura BDI genérica que contém os principais elementos relacionados ao raciocínio prático de agentes racionais limitados. A figura 8.4 apresenta este esquema.

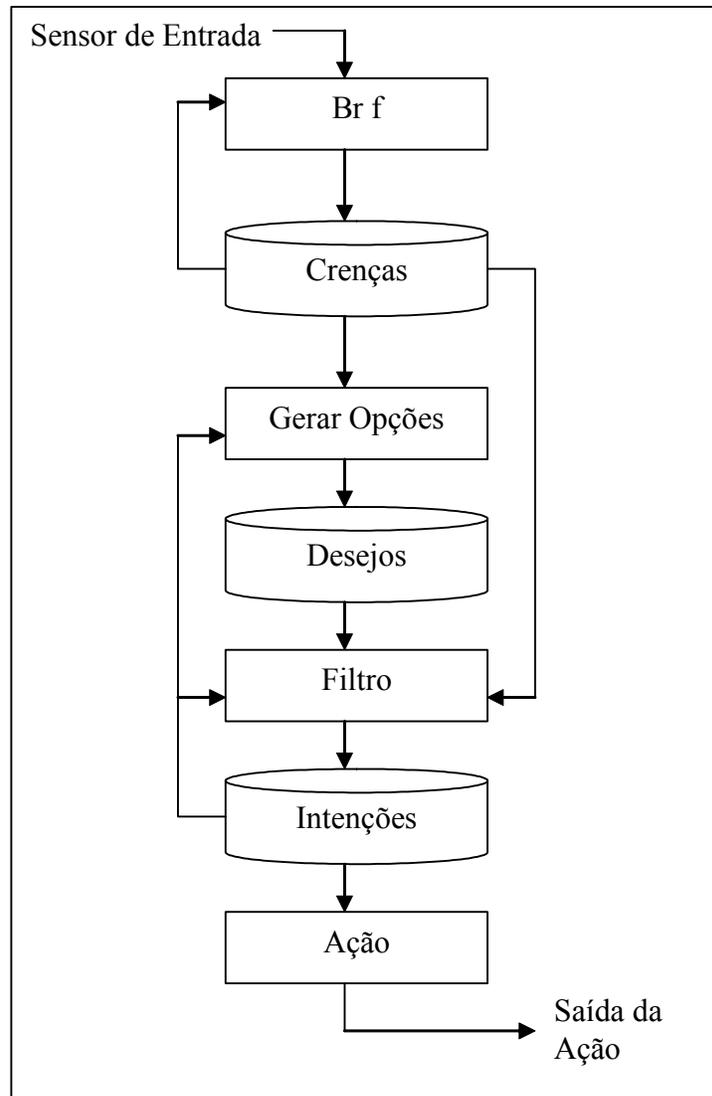


Figura 8.4: Esquema de uma Arquitetura BDI Genérica

Existem sete principais componentes em um agente BDI [WOO 99]:

- um conjunto de crenças atuais, representando informação que o agente tem sobre o seu ambiente corrente;

- uma função de revisão de crenças, ($Br f$), que pega uma entrada perceptual e as crenças correntes do agente, e com base nisto determina um novo conjunto de crenças;
- uma função de geração de opção, ($opções$), que determina as opções disponíveis para o agente (seus desejos), com base nas suas crenças correntes sobre o seu ambiente e suas intenções atuais;
- um conjunto de opções atuais, representando cursos de ação disponíveis para o agente;
- uma função filtro, ($filtro$), que representa o processo de deliberação do agente e que determina as intenções do agente com base nas suas crenças, desejos e intenções atuais;
- um conjunto de intenções atuais, representando o enfoque atual do agente, os estados que ele se comprometeu a alcançar;
- uma função de seleção de ação (execução), que determina uma ação a executar com base nas suas intenções atuais.

O estado de um agente BDI em qualquer momento é uma tripla (B, D, I) onde B é a crença atual do agente e está contido em Bel (conjunto de crenças), D é o desejo atual do agente e está contido em Des (conjunto de desejos) e I é a intenção atual e está contida em Int (conjunto de intenções). A função de revisão de crenças de um agente é um mapeamento $br f: \rho(Bel) \times P \rightarrow \rho(Bel)$, que com base na percepção corrente e nas crenças correntes determina um novo estado de crenças [WOO 2000].

O processo de deliberação do agente BDI começa tentando se identificar quais opções estão disponíveis. Após decidir quais são as opções deve-se escolher entre elas e se comprometer a alguma. As opções escolhidas então se tornam intenções. Deste modo, a função de deliberação pode ser decomposta em dois componentes funcionais:

- geração de opções: nas quais o agente gera um conjunto de alternativas possíveis;
- filtragem: nas quais o agente escolhe entre as alternativas possíveis e se compromete a alcançar algumas alternativas.

A função de geração de opção, $opções$, mapeia um conjunto de crenças e um conjunto de intenções em um conjunto de desejos, $opções: \rho(Bel) \times \rho(Int) \rightarrow \rho(Des)$. Esta função possui diversos papéis. Esta função deve ser responsável pelo raciocínio meio fim, ou seja, o processo de decidir como alcançar as intenções. Logo, após o agente ter formado uma intenção para x , ele deve subsequentemente considerar opção para atingir x . Estas opções serão mais concretas do que x . Como algumas destas opções irão se tornar intenções elas irão fornecer um retorno para a geração de opções, resultando na geração de opções mais concretas. Pode-se pensar no processo de geração de opções de agentes BDI como um processo de recursivamente elaborar uma estrutura de plano hierárquica, considerando e se comprometendo a intenções cada vez mais específicas até que se produzam intenções que correspondem a ações diretamente executáveis.

Além de realizar o raciocínio meio-fim, a função $opções$ deve ser: consistente, ou seja, qualquer opção gerada precisa ser consistente com as crenças e intenções atuais do agente, e oportunista reconhecendo quando as circunstâncias ambientais mudam,

oferecendo aos agentes novas maneiras de atingir intenções, ou possibilitado realizar intenções que antes não eram alcançáveis. A função também deve possibilitar a adoção de novas intenções que podem ser utilizadas para realizar as intenções existentes ou explorar novas oportunidades.

A função filtro decide o que o agente deve fazer, selecionando a melhor opção para o agente se comprometer. A função filtro é representada por $filtro : \rho(Bel) \times \rho(Des) \times \rho(Int) \rightarrow \rho(Int)$. Ela atualiza as intenções dos agentes com base nas suas intenções prévias e nas crenças e desejos atuais. A função filtro deve apresentar as seguintes características [WOO 99]:

- deve desistir de qualquer intenção que não seja mais atingível ou desistir de intenções cujo custo esperado para atingi-las exceda o ganho esperado associado se ela fosse atingida com sucesso;
- deve reter intenções que não foram alcançadas, mas que podem produzir um benefício positivo com o seu sucesso;
- deve adotar novas intenções, ou para atingir as intenções existentes ou para explorar novas oportunidades.

Quando uma opção passa pela função filtro e é escolhida pelo agente como uma intenção, o agente cria um comprometimento com esta intenção. O comprometimento implica em persistência temporal, pois uma intenção, após ser adotada, não deve ser imediatamente descartada. O agente utiliza um mecanismo chamado de estratégia de comprometimento para determinar quando e como desistir de uma intenção. De acordo com Wooldridge [WOO 2000] três estratégias de comprometimento são normalmente discutidas na literatura de agentes racionais:

- comprometimento cego: o agente continua a manter uma intenção até que ele acredite que a intenção foi alcançada;
- comprometimento convergente de direcionamento único: agente continua a manter uma intenção até que ele acredite que a intenção foi alcançada ou que não é mais possível atingir a intenção;
- comprometimento mente-aberta: agente mantém a intenção enquanto ele acredita que ela ainda é possível.

As estratégias de comprometimento estão relacionadas à reconsideração da ação. Desta maneira, o agente irá reconsiderar suas intenções quando uma das três condições acontecerem: ele executou completamente um plano para alcançar suas intenções; ou ele acredita que alcançou suas intenções atuais ou ele acredita que suas intenções atuais não são mais possíveis.

A função *ação* simplesmente retorna de quaisquer intenções executáveis, aquela que corresponde a uma ação diretamente executável, $execução: \rho(Int) \rightarrow A$ [WOO 99].

Wooldridge [WOO 2000] descreve um algoritmo para o laço de controle de um agente BDI. Este laço pode ser visualizado na figura 8.5.

Neste algoritmo Wooldridge considera o raciocínio meio-fim como uma função independente da função *opções*. O raciocínio meio-fim é representado pela função *plano*:

$\rho(Bel) \times \rho(Int) \rightarrow Plan$, que com base nas crenças e intenções determina um plano, do conjunto de todos os planos para alcançar a intenção.

1. enquanto verdade
2. observe o mundo;
3. atualize o modelo interno do mundo;
4. delibere sobre qual a próxima intenção;
5. execute raciocínio meio-fim para escolher um plano para a intenção;
6. execute o plano;
- 7 fim do enquanto

Figura 8.5: Laço de Controle de um Agente BDI

A fim de mostrar a interação das funções descritas, Wooldridge [WOO 2000] apresenta um detalhamento do algoritmo do laço de controle de um agente BDI. Este algoritmo pode ser visualizado na figura 8.6.

1. $B := B_0$;
2. $I := I_0$;
3. enquanto verdade faça
4. pegue a próxima percepção P ;
5. $B := Br f(B, P)$;
6. $D := opções(B, I)$;
7. $I := filtro(B, D, I)$;
8. $\alpha := plano(B, I)$;
9. enquanto não (vazio(α) ou sucesso(I, B) ou impossível(I, B)) faça
10. $\alpha := cabeça(\alpha)$;
11. execute(α);
12. $\alpha := corpo(\alpha)$;
13. pegue a próxima percepção P ;
14. $B := Br f(B, P)$;
15. se reconsiderar(I, B) então
16. $D := opções(B, I)$;
17. $I := filtro(B, D, I)$;
18. fim se
19. se não coerente(α, I, B) então
20. $\alpha := plano(B, I)$;
21. fim se
22. fim enquanto
23. fim enquanto

Legenda:

B : crença

D : desejo

I : intenção

α : plano

cabeça(α): primeira ação do plano

corpo(α): todas as ações de α com exceção da primeira

Figura 8.6: Detalhamento do Algoritmo do Laço de Controle do Agente BDI

Uma questão importante da arquitetura BDI, anteriormente abordada pela arquitetura IRMA e a arquitetura dMARS, é o problema de achar o equilíbrio correto entre estar comprometido com uma intenção ou permitir a reconsideração da intenção. De acordo com Wooldridge [WOO 99] [WOO 2000], o processo de deliberação deve estar afinado com o seu ambiente, garantindo que em ambientes mais dinâmicos e altamente imprevisíveis o agente reconsidere suas intenções frequentemente e, por outro lado, em ambientes mais estáticos a reconsideração seja menos freqüente. Isto pode ser realizado no esquema apresentado por Wooldridge através da função reconsiderar presente no algoritmo do laço de controle (figura 8.6). Desta maneira, Wooldridge apresenta uma arquitetura de agente BDI genérico que abrange todos os conceitos relacionados ao raciocínio prático que compõem o modelo BDI.

8.2.4 Arquitetura BDI para Intenções Conjuntas

Jennings [JEN 93a] aborda no seu trabalho a questão de um modelo de colaboração usado para guiar o comportamento de uma sociedade de agentes racionais. Como os agentes são racionais, Jennings assume a existência de intenções, crenças e desejos. Contudo, Jennings considera que as intenções como são tradicionalmente abordadas na arquitetura BDI são insuficientes para tratar o aspecto da colaboração de agentes, pois as intenções definem somente o comportamento individual do agente.

Jennings [JEN 93b] entende que existem duas principais limitações com a abordagem da intenção individual: primeira, a ação conjunta é mais do que a soma das ações individuais, mesmo se as ações são coordenadas; segunda, existe uma diferença fundamental entre indivíduos e grupos. Esta diferença pode ser vista mais claramente se considerando a noção de comprometimento. O comprometimento do grupo não pode ser simplesmente uma versão de comprometimentos individuais onde um grupo é o agente, porque os membros podem divergir nas suas crenças e um agente individual não pode. Se um indivíduo passa a acreditar que um objetivo é impossível, então é racional para o agente desistir do objetivo. Da mesma maneira, quando o objetivo do grupo como um todo é impossível, o grupo deve parar de tentar realizar aquele objetivo. No primeiro caso, o indivíduo pode desistir do objetivo porque ele possui conhecimento suficiente para isto. Entretanto, no segundo caso, todos os componentes do grupo podem não ter conhecimento suficiente para realizar isto.

Para superar as limitações das intenções individuais, são investigadas as intenções conjuntas. De acordo com Jennings [JEN 93a], intenções conjuntas podem ser intuitivamente definidas como comprometimentos conjuntos para executar uma ação coletiva enquanto os agentes estão em um determinado estado mental compartilhado. Neste caso, a colaboração é diretamente dependente dos estados mentais dos participantes. Desta maneira, dois grupos de agentes poderiam estar executando exatamente as mesmas ações, um grupo poderia estar agindo colaborativamente, se eles compartilham os estados mentais necessários, o outro grupo não.

Se os agentes querem possuir intenções compartilhadas eles devem:

- concordar em um objetivo comum, compartilhado;

- concordar que eles desejam colaborar para alcançar seus objetivos compartilhados;
- concordar em um meio comum (plano) para atingir seus objetivos;
- reconhecer que as intenções executadas por diferentes agentes estão relacionadas;
- ter critério para rastrear a racionalidade dos seus comprometimentos;
- ter regras de comportamento que definem como se comportar localmente e em relação a outros quando a ação conjunta está progredindo como esperado e quando ela apresenta dificuldades.

A responsabilidade conjunta tem como objetivo ditar as regras de comportamento que definem como se comportar localmente e em relação a outros agentes. Um agente deve permanecer comprometido a alcançar um objetivo comum, considerando o acordo que foi realizado com os outros agentes para a solução de um problema. Este tipo de responsabilidade está relacionado ao comprometimento individual do agente para com a solução. Um agente irá deixar este comprometimento quando: a saída desejada de um passo do plano já está disponível; a execução da seqüência de ações não atinge as conseqüências desejadas; uma das ações especificadas não pode ser executada; ou uma das ações não foi executada.

Quando um dos membros do grupo acredita que o processamento não é apropriado, porque ele tem conhecimento que não é mais racional se comprometer a um objetivo comum ou aos meios de executá-los, o agente deve disseminar esta informação o mais amplamente possível. Desta maneira, o modelo de responsabilidade estimula que quando um membro do grupo não está mais comprometido com uma ação conjunta ele deve garantir que seus companheiros acreditem que ele não está mais comprometido e o porquê de ele não estar mais comprometido. Isto possibilita que o grupo faça uma nova avaliação da viabilidade da ação conjunta e em particular das ações envolvendo o agente que não está mais comprometido. Se uma ação conjunta precisa ser abandonada ou refinada, a quantidade de recursos gastos da comunidade é mantida no mínimo, pois as atividades que não fazem mais sentido são interrompidas na primeira oportunidade.

De acordo com Jennings [JEN 93a], a responsabilidade conjunta define o papel que as intenções individuais e conjuntas têm no processo de solução de problemas colaborativos. Elas são usadas para coordenar ações (intenções dirigidas ao futuro) e controlar a execução das ações atuais (intenções dirigidas ao presente).

Considerando as idéias de responsabilidade conjunta, Jennings [JEN 93a] apresenta uma arquitetura de alto nível para agentes racionais baseada no modelo de crenças-desejos e intenções conjuntas (figura 8.7) para intenções dirigidas ao futuro.

A arquitetura (figura 8.7) mostra que o agente possui duas fontes de eventos: os que ocorrem localmente e os que ocorrem na comunidade. Quando um evento é detectado pelo monitor de eventos ele gera um novo objetivo que serve de entrada para a análise meio-fim. A análise meio-fim deve determinar se o objetivo deve ser alcançado e, em caso positivo, qual a melhor maneira de se atingir este objetivo. Para determinar se um objetivo deve ser alcançado a análise meio-fim deve levar em consideração sua biblioteca de receitas e suas intenções atuais. A biblioteca restringe os cursos de ação disponíveis e indica o tipo de atividade necessária para a realização destes cursos de ação (atividade local, atividade

social ou uma combinação de atividades locais e sociais). As intenções devem ser levadas em consideração, pois elas indicam as atividades para as quais o agente já está comprometido. A saída da análise meio-fim é uma decisão que indica se o objetivo deve ser atingido ou não, e se for, se o objetivo vai ser realizado localmente ou socialmente.

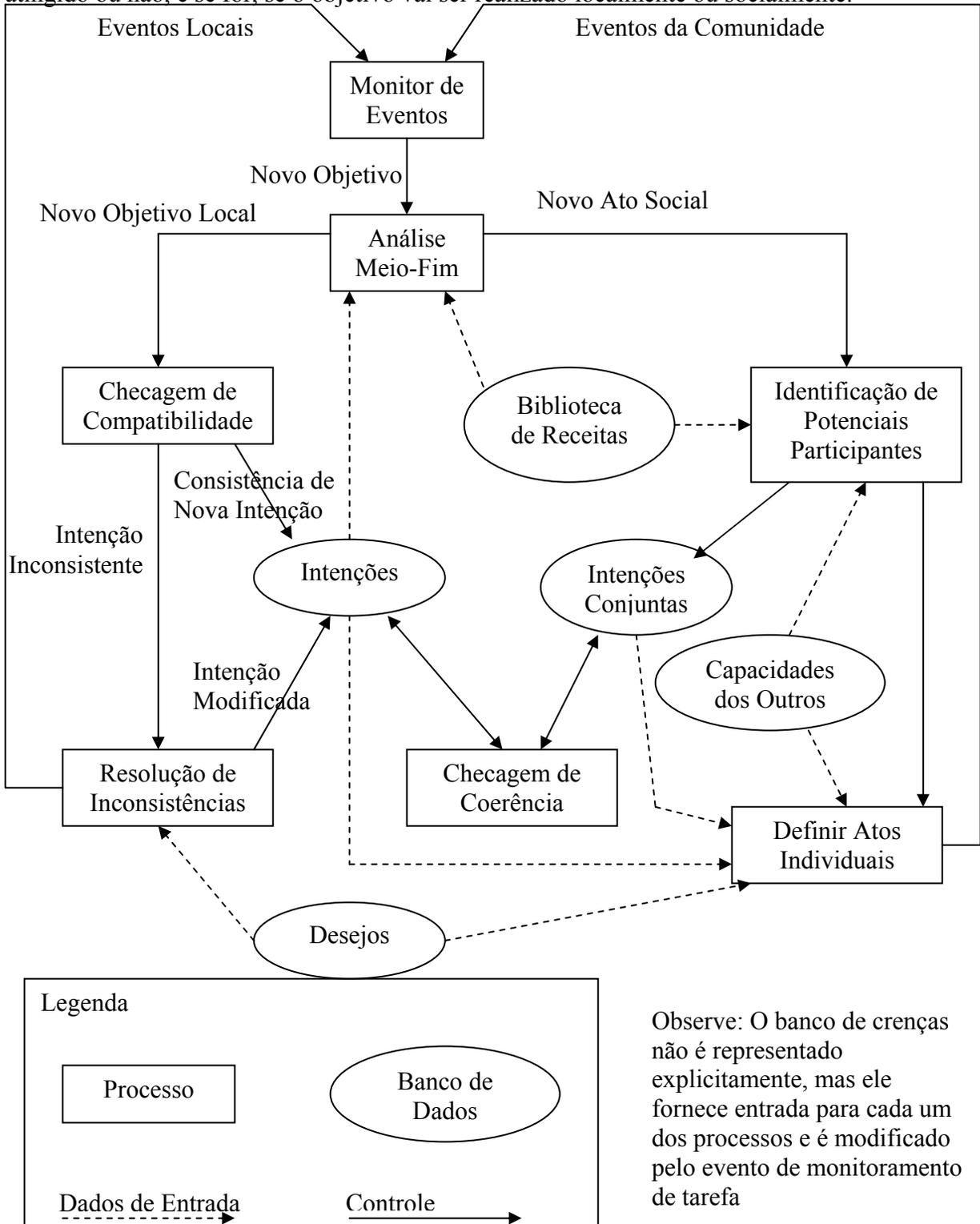


Figura 8.7: Arquitetura Funcional do Agente – Intenções Dirigidas ao Futuro

Se o agente decide realizar o objetivo localmente, ele deve garantir que a nova intenção é compatível com as intenções existentes. Caso o objetivo esteja em conflito com as intenções existentes, a inconsistência deve ser resolvida. A inconsistência pode ser resolvida modificando-se os comprometimentos existentes ou alterando o novo objetivo para que ele não provoque mais o conflito. A escolha de qual técnica utilizar irá depender dos desejos do agente. Se o novo objetivo é menos importante, ou seja, menos desejável do que os existentes, então ele deve ser alterado, caso contrário, os objetivos existentes é que devem ser alterados. Qualquer modificação realizada pelo mecanismo de resolução de inconsistências deve refletir na representação da intenção.

Se o agente decidir realizar o objetivo colaborativamente, deve-se estabelecer uma ação social. O primeiro passo é identificar quais agentes da sociedade estão interessados em participar desta colaboração. Isto pode ser feito de diferentes maneiras, o agente pode ditar quem deve participar ou pode ocorrer um processo de negociação entre os agentes envolvidos. Para estabelecer uma ação social, o agente deve levar em consideração os seguintes fatores: o tipo de organização da sociedade (hierárquica ou não), a estratégia de formação do grupo (grupos pequenos ou grandes) e conhecimento das capacidades dos outros membros da sociedade. Após os participantes terem sido identificados, é criado um esquema de intenção conjunta. Contudo, este esquema somente afirma o interesse dos participantes, ele ainda não contém o grupo final que irá executar a atividade colaborativa nem a solução que vai ser realizada.

As intenções conjuntas não podem ser executadas diretamente, elas servem como uma descrição que relaciona ações a múltiplos agentes. Somente os participantes do grupo podem agir e conseqüentemente as intenções individuais dos participantes estão diretamente relacionadas às ações. Contudo, cada participante deve adotar pelo menos uma intenção individual como conseqüência da sua participação na ação conjunta. Deste modo, o agente não pode estar comprometido com uma ação na intenção conjunta que seja incompatível com algumas das suas intenções individuais. A consistência entre as intenções individuais e as intenções conjuntas é realizada pela checagem de coerência, que é chamada cada vez que uma nova proposta de ação conjunta ou individual é realizada.

O relacionamento das ações a serem executadas com os participantes disponíveis é guiado pelos fatores de estratégia, organização desejada e conhecimento das capacidades dos agentes, como mencionado anteriormente. A maneira como o agente irá executar suas ações individuais é deixada em aberto para que o agente escolha a melhor opção de acordo com a situação em que ele se encontra. A saída deste processo é enviada para cada participante, e cada um deles deve realizar um processo de análise meio-fim para relacionar as ações da ação conjunta com os seus comprometimentos existentes. Qualquer inconsistência deve ser identificada pela checagem de consistência e pode levar a uma nova reconsideração do mapeamento das intenções conjuntas para as intenções individuais.

Além de fornecer mecanismos para coordenar ações, as intenções também guiam a execução e monitoramento das ações (figura 8.8).

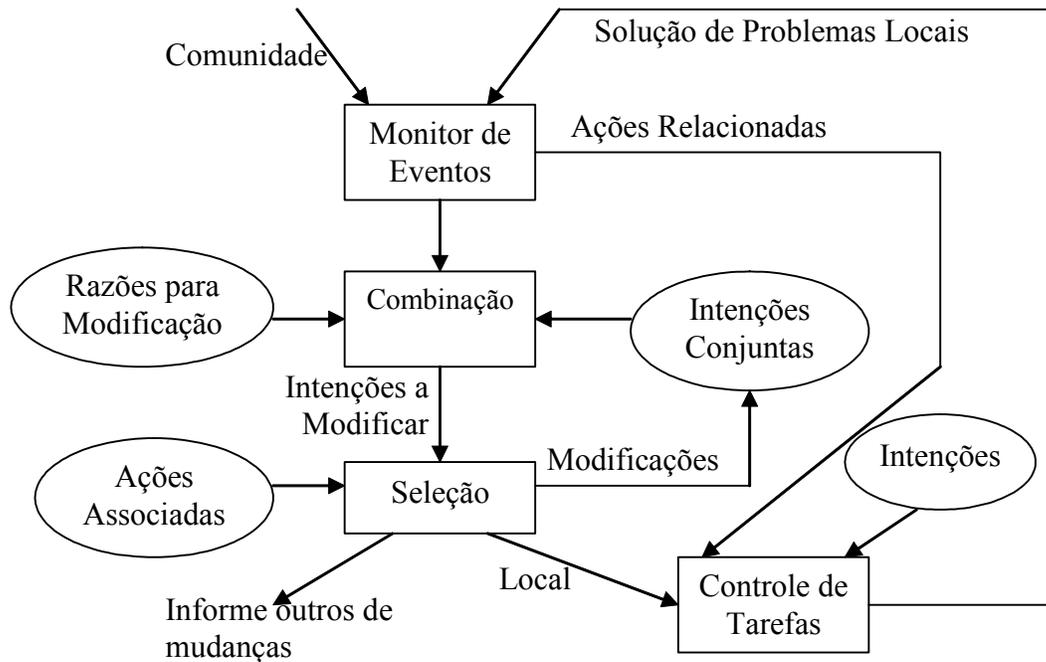


Figura 8.8: Arquitetura Funcional do Agente – Intenções dirigidas ao Presente

A arquitetura apresentada na figura 8.8 mostra que as intenções são usadas para controlar a execução das ações e monitorar os comprometerimentos para garantir que eles ainda sejam racionais. Quem garante que os comprometerimentos sejam honrados é o controle de tarefas. Desta maneira, este processo pode suspender ou interromper tarefas a fim de garantir que o agente cumpra com seus comprometerimentos. O controle de tarefas é informado quando ações, locais ou de outros participantes, relacionadas aos comprometerimentos são realizadas. Isto proporciona o sincronismo das ações do controle de tarefas.

Para monitorar os comprometerimentos, a arquitetura especifica as razões pelas quais uma ação conjunta deve ser modificada ou finalizada e as ações que devem ser tomadas quando isto acontece. O componente razões para a modificação define as condições nas quais as ações conjuntas devem ser finalizadas ou novamente analisadas. Após o processo de combinação ter identificado as intenções conjuntas que precisam ser analisadas novamente, o agente deve decidir qual ação deve ser realizada para remediar o problema. O modelo de ações associadas descreve atividades apropriadas para várias classes de eventos. Podem existir vários tipos de eventos que levam um comprometerimento ser desfeito. Se o comprometerimento ao objetivo geral é desfeito, o agente deve parar de executar todas as suas ações atuais e planejadas e garantir que os outros participantes sejam informados imediatamente.

Jennings [JEN 93a] mapeou as arquiteturas funcionais apresentadas anteriormente em uma arquitetura de alto-nível chamada de GRATE*, utilizada para o domínio de gerenciamento de transporte de energia. A arquitetura funcional apresentada por Jennings possibilitou a extensão da idéia tradicional de intenções. As intenções foram ampliadas para intenções conjuntas, possibilitando aos agentes usar um modelo de agência racional BDI para a realização de atividades de colaboração.

8.2.5 Arquitetura BDI com Restrições

Chalmers [CHA 2001] pesquisa o uso de restrições dentro de uma ferramenta BDI. O uso de restrições permite expressar os desejos de um agente como restrições complexas quantificadas e modelar o processo de tomada de decisão como um programa de restrição lógica. Chalmers acredita que o uso de restrições no processo de deliberação e a representação dos desejos dos agentes como restrições fornecem uma maneira flexível de dar autonomia suficiente ao agente para ele tratar com situações inesperadas e com informação incompleta do ambiente.

Chalmers [CHA 2001] apresenta uma arquitetura de agentes BDI (figura 8.9) dividida em dois níveis distintos: nível de coordenação e escalonamento e nível BDI. O nível de coordenação e escalonamento controla as tarefas do agente em alto nível e coordena a comunicação e interação do agente com os outros e com o ambiente. A camada BDI constrói e executa planos para tarefas individuais.

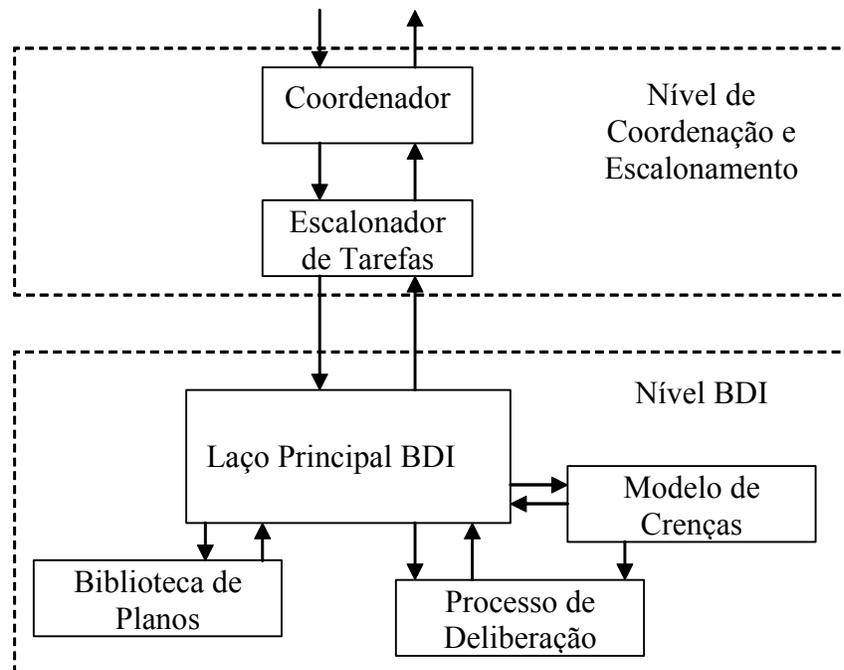


Figura 8.9: Arquitetura do Agente

O escalonador é um programa baseado em restrições que pega um determinado conjunto de tarefas (como restrições) e tenta encontrar um escalonamento viável que permita ao agente completar o mais amplo (ou mais importante) conjunto de tarefas. Isto também possibilita ao agente encontrar qualquer tarefa que não possa ser completada e que, desta maneira, é candidata para delegação ou troca. As tarefas representam os desejos do agente e são representadas como restrições. Esta representação fornece uma linguagem comum sob a qual as tarefas podem ser executadas e entendidas entre agente que compartilham um mesmo modelo de crenças.

A informação é passada entre as duas camadas à medida que o agente progride no escalonamento e execução de tarefas. A camada de coordenação e escalonamento fornece à

camada BDI a tarefa atual a ser executada como uma restrição de alto nível. A camada BDI escolhe e executa uma ação ou subplano correspondente. A informação que flui do sentido contrário fornece detalhes sobre a execução do plano corrente (tal como qual subplano ou ação o agente se comprometeu, estimativas de término de termo para a tarefa). Usando esta informação a camada de coordenação e escalonamento pode construir um escalonamento mais detalhado que irá levar em conta comprometimentos que o agente tem devido a ações prévias.

A camada BDI da arquitetura do agente possui crenças expressas como fatos, desejos representados como restrições, intenções representadas como estruturas Prolog e a biblioteca de planos representada como uma coleção de planos STRIPS. Chalmers baseia seu modelo de agentes BDI no modelo de Rao e Georgeff [RAO 95]. A figura 8.10 mostra a versão em pseudocódigo do algoritmo do laço principal do agente BDI. Este algoritmo controla o planejamento do agente e sua interação com o ambiente.

```

Faça
  opções:= gere-opção(fila_eventos, B, D, I)
  opções-selecionadas := delibere(options, B, D, I)
  atualize_intenções(opções-selecionadas, I)
  execute(I)
  pegue-novos-eventos-externos();
  libere-atitudes-que-tiveram-sucesso(B, D, I);
fim

```

Figura 8.10: Laço Principal BDI [CHA 2001]

Durante o processo de deliberação, o agente deve ser capaz de escolher seu próximo curso de ação. Ele deve deliberar sobre planos disponíveis e decidir o mais apropriado baseado em alguns fatores. Estes fatores são: o estado atual da tarefa sendo executada; as crenças dos agentes sobre o ambiente atual; o prazo da tarefa que está atualmente sendo executada e outras tarefas que o agente tem programado para fazer a seguir.

As crenças do agente sobre o estado do ambiente, as tarefas que o agente tem que realizar (seus desejos) e o estado atual da sua tarefa (intenção) são usados como base para o processo de deliberação. As crenças são um reflexo de como o agente vê o estado atual do mundo. Deste modo, elas naturalmente formam a base para o processo de deliberação do agente. Os desejos dos agentes são formulados em termos de entidades, atributos e relacionamentos contidos no modelo de crenças. Os autores usam estes desejos como especificações de tarefas necessárias.

Quanto o agente inicia o seu processo de deliberação, ele constrói um banco de dados de soluções que irá conter todas as soluções possíveis. Para colocar todas as soluções possíveis neste banco de dados, o agente encontra todas as ações possíveis do estado atual e as executa, armazenando os estados resultantes no banco de dados da solução. Depois o agente pega os estados que estão no banco de dados de solução e encontra as ações disponíveis para cada um deles. Estas ações são executadas e se encontram novos estados. Este processo é repetido até uma determinada profundidade. Desta maneira, o banco de dados de soluções resultantes contém todos os estados possíveis das crenças dos agentes para os próximo n passos especificados. Cada candidato no banco de dados de solução

possui uma ligação para o estado que o segue. Assim, os autores conseguem uma estrutura semelhante a de mundos possíveis que preserva a informação sobre a posição da hierarquia de cada estado em relação aos outros [CHA 2001].

Para solucionar o processo de deliberação como um CLP (problema de restrição lógica), todos os candidatos do banco de dados de solução são retornados e transformados em domínios finitos no *ECLipSe*, um resolvidor de restrições baseado em Prolog. As restrições então são colocadas contra estes domínios finitos e qualquer solução inválida é eliminada.

A comunicação entre os agentes acontece quando um único agente não pode completar um conjunto de tarefas dadas e requer a assistência de outros agentes para completar uma ou mais destas tarefas. O agente deve descobrir qual, ou quais tarefas, ele não pode completar, criando, na medida em que planeja e executa, um escalonamento que mostra o início e o tempo esperado de término da atividade atual e das atividades futuras. Este escalonamento deve ser capaz de ressaltar tarefas problemas, que irão formar a base da comunicação do agente e coordenação com outros. O escalonador pega os desejos do agente e tenta encontrar um escalonamento apropriado para atingir estes desejos dadas as restrições especificadas. Estas restrições podem vir de várias fontes como: a prioridade de uma tarefa definida pelo usuário; a localização da tarefa em relação a localização do agente; o tempo estimado para a finalização da tarefa e a precedência das tarefas.

Cada tarefa é representada como um domínio finito de pontos de partida sob as quais o escalonador tenta construir um escalonamento. Se o escalonador não encontra nenhum escalonamento que possibilite o término de cada tarefa, então o escalonamento que contém a menor quantidade de sobreposição entre as tarefas é retornado. Este escalonamento é usado como base para a seleção de tarefas para delegar e trocar durante a coordenação entre os agentes.

O escalonador primeiro pega todas as tarefas e encontra, da informação contida nas crenças dos agentes, a duração estimada de cada tarefa. No início do escalonamento, o agente possui um mapa de cenário de teste que mostra a posição e as conexões nos principais pontos. Da informação obtida por este mapa, o agente constrói uma lista mostrando as distâncias entre a posição de início de cada tarefa e a localização atual do agente. O princípio utilizado é de que o agente irá tentar realizar primeiro a tarefa que está mais próxima dele. A próxima tarefa escolhida é aquela cujo ponto de início é o mais próximo do ponto de fim da anterior. Dada a informação inicial de escalonamento, o escalonador utiliza um resolvidor de restrições para fornecer o melhor escalonamento baseado nos tempos de término estimados para as tarefas.

A principal tarefa do escalonador é encontrar tarefas problema, ou seja, tarefas que não podem ser completadas junto com outras. O escalonador faz isto encontrando as tarefas que se sobrepõem com um maior número de outras tarefas. O escalonador deve sugerir a delegação desta tarefa, pois a remoção desta tarefa irá possibilitar a realização de um maior número de tarefas.

Chalmers [CHA 2001] salienta que a idéia de combinar um modelo de dados (para representar as crenças) e restrições (para representar desejos e influenciar o processo de deliberação) fornece uma maneira efetiva de se raciocinar e se comunicar em ambientes dinâmicos e desconhecidos. O uso de restrições na construção da arquitetura de agentes

BDI coloca no agente uma maneira flexível de representar e manipular tarefas e conhecimento enquanto preserva a autonomia do agente.

8.3 Lógicas BDI

Em paralelo aos esforços de especificar arquiteturas que suportem a idéia do raciocínio prático, alguns trabalhos, como [RAO 91a] [RAO 91b] [KON 93] [CAV 95a] [CAV 95b], têm sido desenvolvidos no sentido de fornecer uma formalização lógica para o modelo BDI. Os primeiros trabalhos em lógica BDI foram desenvolvidos por Rao e Georgeff [RAO 91a] [RAO 91b] e formalizaram os conceitos de crenças, objetivos, intenções, ações e os inter-relacionamentos entre eles. Para Rao e Georgeff estes conceitos podem ser definidos através de uma lógica de mundos possíveis ramificada no tempo.

A lógica de mundos possíveis modela o mundo usando uma estrutura temporal chamada de árvore de tempo que possui um futuro especificado através de ramos e um único passado. Um ponto particular neste mundo é chamado de situação. Para se ir de um ponto para outro no mundo é necessário que ocorram eventos. Existem dois tipos de eventos possíveis: primitivos e não primitivos. Os eventos primitivos são eventos que podem ser executados diretamente pelo agente e determinam o próximo ponto na árvore de tempo. Eventos não-primitivos mapeiam pontos não adjacentes, possibilitando modelar a natureza parcial dos planos.

Os ramos na árvore de tempo representam as escolhas disponíveis para o agente em cada momento do tempo. Se ao executar um evento, um determinado ramo tem sucesso, ele é nomeado com sucesso. Caso contrário, se o ramo falha, ele é nomeado com falha.

Cada situação, ou seja, cada ponto particular do mundo é associado a um conjunto de mundos acessíveis às crenças. Estes mundos são os mundos que o agente acredita serem possíveis. Ao contrário das representações tradicionais de modelo de crenças, cada mundo de crenças é uma árvore de tempo. A existência de vários mundos acessíveis a crenças é uma indicação de falta de conhecimento sobre o estado do mundo. Contudo, dentro destes mundos as ramificações futuras indicam as escolhas que ainda estão disponíveis para o agente no momento de selecionar suas ações para executar.

Da mesma maneira que as crenças, para cada situação é associado um conjunto de mundos acessíveis a objetivos para representar os objetivos do agente. Rao e Georgeff [RAO 91a] entendem objetivos como desejos escolhidos pelo agente e estes objetivos são consistentes uns com os outros. O agente também deve acreditar que o objetivo é alcançável. Isto previne o agente de adotar objetivos que ele acredita não serem possíveis.

As intenções também são representadas através de mundos possíveis. Os mundos acessíveis às crenças são os mundos que o agente se comprometeu a tentar realizar. Da mesma maneira que existe compatibilidade dos objetivos com as crenças, os mundos das intenções devem ser compatíveis com os mundos dos objetivos, ou seja, o agente somente pode ter uma intenção se esta intenção é um dos seus objetivos.

Enquanto para cada mundo acessível às crenças deve existir um mundo acessível a objetivos (e o mesmo para intenções) o contrário não precisa necessariamente acontecer. Desta maneira, mesmo que o agente acredite que certos fatos são inevitáveis, ele não precisa adotá-los como objetivos. Isto significa que, embora os objetivos e as intenções

tenham que ser consistentes com as crenças, elas não precisam englobar todas as crenças do agente.

A semântica das crenças, objetivos e intenções é formalizada em uma extensão da lógica proposicional CTL* [EME 89]. É descrita uma variação da lógica e são introduzidos operadores para crenças, objetivos e intenções. Em [RAO 91b] é apresentada uma semântica expressiva para intenções e são examinados os axiomas que fazem o relacionamento entre intenções e crenças e intenções e objetivos.

Podemos resumir a estrutura da lógica de Rao e Georgeff da seguinte maneira: cada mundo é uma estrutura temporal com uma ramificação de tempo futuro e um único passado, chamada de árvore de tempo. Um ponto particular do tempo em um mundo específico é chamado de situação. Eventos transformam um ponto do tempo em outro ponto. Cada situação está associada com um conjunto de mundos acessíveis a crenças, objetivos e intenções. Isto representa os mundos que o agente acredita serem possíveis, os desejos que ele possui sobre o mundo e seus comprometimentos para alcançar seus desejos. Múltiplos mundos possíveis resultam de uma falta de conhecimento do agente sobre o estado do mundo. Dentro de cada um destes mundos possíveis, as ramificações futuras representam as escolhas de ações disponíveis para o agente. A movimentação do mundo das crenças para o mundo dos objetivos e para o mundo das intenções pode sucessivamente os ramos da árvore de tempo, realizando escolhas seletivas sobre as ações futuras do agente.

Rao e Georgeff [RAO 91a] estendem o poder expressivo da lógica de mundos possíveis para modelar o processo de deliberação pela introdução de probabilidades e benefícios. Esta extensão funciona da seguinte maneira: um agente tem em cada situação uma distribuição de probabilidade nos seus mundos acessíveis às crenças. O agente escolhe submundos que ele considera valer a pena examinar e associa um valor de benefício com cada caminho neste submundo. Estes submundos então são considerados os mundos acessíveis a objetivos. Fazendo uso da distribuição de probabilidade sobre os mundos acessíveis às crenças e a distribuição de benefícios nos caminhos dos mundos acessíveis a objetivos, o agente determina o(s) melhor(es) plano(s) de ação para diferentes cenários. Este processo é a deliberação dos mundos possíveis e tem como resultado um conjunto de submundos de mundos acessíveis aos objetivos que o agente considera mais desejáveis. Estes submundos são os mundos acessíveis às intenções que o agente se compromete a alcançar. A deliberação é feita através de uma função de deliberação que considera as probabilidades e benefícios para escolher um ou mais cursos de ação para executar em um determinado momento [RAO 95].

Os trabalhos de Rao e Georgeff examinam aspectos filosóficos importantes como: o tratamento de crenças; objetivos e intenções; a idéia de intenções como parte das crenças e dos objetivos dos agentes e como a deliberação leva a formação de intenções. Todos estes aspectos são abordados através do modelo de mundos possíveis, formalizados através de uma extensão da lógica proposicional CTL*.

8.4 Considerações sobre o Capítulo

Neste capítulo apresentamos os três elementos importantes do modelo BDI, como indicado por Wooldridge [WOO 2000]: teoria da ação racional, arquiteturas BDI e formalização de lógicas BDI.

A teoria da ação racional ou raciocínio prático, aponta para o papel fundamental das intenções como condutores das ações futuras dos agentes. As intenções dirigem o raciocínio meio-fim, restringem a deliberação futura através da sua característica de persistência e influenciam as crenças sobre as quais o raciocínio futuro está baseado. As intenções estão intimamente relacionadas ao planejamento, na medida em que são elementos dentro de planos maiores. Os planos contêm cursos de ação abstratos que o agente se compromete a executar. Os planos não ditam completamente as ações dos agentes, sendo planos parciais que devem ser completados nos momentos adequados. Desta maneira, Bratman [BRA 88] [BRA 99a] compartilha da idéia de Agre e Chapman [AGR 88a] [AGR 88b] de que planos devem ser estruturados como intenções e não como programas. Assim, planos são um comprometimento a um objetivo que guiam as ações dos agentes, mas não determinam suas ações específicas.

As arquiteturas BDI apresentadas na seção 8.2 são baseadas no modelo de raciocínio prático de Bratman e desta maneira representam os conceitos de crenças, desejos (objetivos), intenções, planos, raciocínio meio-fim, deliberação e comprometimento. Contudo, a arquitetura dMARS entende planos como programas, pois indica que seus planos são procedurais e que podem ser diretamente executados pelos seus agentes. Esta idéia é contrária à idéia de Bratman de ver planos como intenções e não planos como programas.

De acordo com Pollack [POL 92][POL 99] e Bratman [BRA 99a], os agentes BDI são agentes planejadores, mas não são planejadores no sentido de produzir um plano completo de ação, pois é impossível prever todas as situações que o agente pode enfrentar no futuro. Desta maneira, a visão de planos como intenções já implica que os agentes não são planejadores no sentido tradicional, mas sim deixam opções abertas para serem completadas no momento adequado. Entendemos que esta característica está relacionada a agentes improvisacionais e propomos que os agentes BDI sejam entendidos como agentes improvisacionais implícitos e não como agentes de planejamento. No próximo capítulo abordaremos mais profundamente esta questão.

Por último, as formalizações lógicas iniciadas por Rao e Georgeff [RAO 91a], e seguida por outros autores, são importantes para fornecer uma descrição semântica formal das questões envolvidas no modelo BDI.

No próximo capítulo iremos analisar as arquiteturas BDI apresentadas na seção 8.2 procurando identificar os tipos de improvisação, descritos no capítulo 6, presentes implicitamente nestas arquiteturas.

9 IDENTIFICAÇÃO DE TIPOS DE IMPROVISACÃO IMPLÍCITA NAS ARQUITETURAS BDI

Neste capítulo iremos identificar a existência implícita de diferentes tipos de improvisação nas arquiteturas BDI apresentadas no capítulo 8. Por improvisação implícita, entendemos aquela improvisação que não foi planejada e pensada para estar na arquitetura, mas que foi incluída instintivamente pelos projetistas. Iremos considerar e avaliar a improvisação que ocorre no momento de construir as ações dos agentes. É importante salientar que, como os trabalhos analisados não enfocam as questões de personalidade e interação com o usuário, os tipos de improvisação de personalidade e público não podem ser analisados. Contudo, podemos tirar algumas conclusões em relação à improvisação implícita de público em dois trabalhos, na arquitetura IRMA e no diagrama esquemático de Wooldridge.

Desta maneira, os tipos de improvisação implícita que iremos examinar nas arquiteturas BDI são: improvisação de texto e improvisação de solução de problemas. Embora não possamos analisar a improvisação de personalidade, poderíamos considerar que a mesma possa ser executada de maneira análoga à construção das ações do texto.

9.1 Arquitetura IRMA

Nesta seção iremos analisar onde ocorrem implicitamente os tipos de improvisação na arquitetura IRMA. Maiores detalhes sobre a arquitetura IRMA no capítulo 8, seção 8.2.1.

9.1.1 Improvisação de Texto

9.1.1.1 Fase de Preparação

Na fase de preparação do texto tanto o raciocínio meio-fim, o mecanismo de filtragem, composto pelo mecanismo de compatibilidade e mecanismo de reescrita, quanto o processo de deliberação estão envolvidos na construção da intenção e estão relacionados à produção de alternativas de cursos de execução para o agente. Os processos de raciocínio meio-fim, de filtragem e de deliberação manipulam e especificam opções de ação que definem cursos de ação alternativos para os agentes.

Os artigos da arquitetura IRMA não deixam claro como é realizada a escolha da ação em função da intenção, caso o processo de deliberação indique a seleção de mais de uma opção possível para fazer parte da intenção do agente.

9.1.1.2 Fase de Execução

Embora os artigos não especificuem claramente como é feita a escolha de uma opção a ser executada em função da intenção em um determinado momento, a figura 8.1 e 8.2 do capítulo 8, seção 8.2.1 mostram a ação resultando da estrutura da intenção. Isto indica que de alguma forma o agente escolhe uma opção de ação em função da sua intenção.

9.1.2 Improvisação de Público

A arquitetura IRMA não possui representação das improvisações dos usuários. Contudo, considerando que os usuários poderiam provocar efeitos no ambiente e que estes efeitos podem ser monitorados através da análise perceptual do ambiente, a arquitetura consegue analisar novas opções geradas.

9.1.3 Improvisação de Resolução de Problemas

Não existe, porque considera somente os eventos ambientais para os quais o agente já possui uma sensibilidade. Seria como uma sensibilidade a uma oportunidade previamente conhecida, no sentido de já ter sido previamente incorporada nas possíveis improvisações do texto. A questão que se pretende enfrentar, nesta tese, é a ocorrência de um evento ou situação que aconteceu ao acaso e para o qual o agente não possui uma seqüência de ações apropriada.

As questões das oportunidades são tratadas na arquitetura, apresentada na figura 8.1, através do analisador de oportunidade. O analisador de oportunidade foi transformado na arquitetura IRMA no mecanismo de reescrita que é responsável por receber as “sensibilidades a oportunidades” da análise perceptual. Contudo, são oportunidades previamente consideradas pelo agente como possíveis de acontecer em um determinado momento.

9.2 Arquitetura dMARS

Nesta seção iremos apresentar os tipos de improvisação que ocorrem implicitamente na arquitetura dMARS. Maiores informações sobre esta arquitetura se encontram no capítulo 8, seção 8.2.2.

9.2.1 Improvisação do Texto

No dMARS a fase de preparação e execução da improvisação do texto é realizada por um único módulo, o interpretador. Neste caso, as duas fases estão combinadas.

O interpretador é responsável por gerenciar todas as operações dos agentes dMARS. Através do seu ciclo de execução, ele encontra planos cuja condição de ativação combine

com um evento da lista de eventos. O interpretador seleciona deste conjunto um plano para execução e coloca na pilha de intenções. Depois pega o plano que está no topo da pilha e o executa.

Um plano pode ser composto de ações diretamente executadas ou por novos planos. O interpretador realiza um planejamento de subobjetivos sensíveis ao contexto, que é deferido até o ponto no qual o subobjetivo é selecionado para execução.

Dentro da nossa visão de que planejamento e improvisação são extremos de uma atuação que pode ser mais ou menos planejada ou improvisada, entendemos que a improvisação do texto no caso da arquitetura dMARS está mais para planejamento do que para improvisação. Pode existir alguma improvisação no sentido de que existem vários planos possíveis que combinam com um evento, contudo o interpretador do agente escolhe somente um deles para prosseguir. O mais apropriado seria chamar este processo de planejamento improvisado do texto e não improvisação do texto, pois a ênfase está no planejamento e não na improvisação.

9.2.2 Improvisação de Resolução de Problemas

Durante a procura por um plano a ser executado na arquitetura dMARS, se um caminho falha e não existem mais caminhos possíveis, o plano falha. Quando o plano é o único da pilha, a intenção falha completamente. A arquitetura não aborda uma solução para o problema de não se ter mais alternativas para o plano e também não é mencionado o que ocorre quando um determinado evento não dispara nenhum plano, ou seja, não existe plano na biblioteca de planos capazes de resolver o problema imediato.

9.3 Diagrama Esquemático de Wooldridge para Arquitetura BDI Genérica

Nesta seção apresentaremos a existência dos tipos de improvisação implícita no diagrama esquemático de Wooldridge. Maiores informações sobre esta arquitetura podem ser obtidas no capítulo 8, seção 8.2.3.

9.3.1 Improvisação de Texto

9.3.1.1 Fase de Preparação

A improvisação de texto na fase de preparação está dividida em duas funções do diagrama esquemático, a função de geração de opções e a função filtro. A função de geração de opções é responsável por determinar as opções disponíveis para o agente considerando suas crenças sobre o ambiente e suas intenções atuais. A função filtro recebe as opções disponíveis, ou seja, os desejos, e delibera sobre quais destas opções se transformarão em intenções futuras dos agentes, sendo estas intenções vistas como um curso abstrato de ação, que não define a ação no seu nível primitivo.

As funções presentes na fase de preparação envolvem a construção da intenção fornecendo opções alternativas para futura execução.

9.3.1.2 *Fase de Execução*

A improvisação do texto na fase de execução ocorre na função de seleção da ação. Esta função seleciona dentro das ações disponíveis quais são mais apropriadas considerando a intenção atual.

9.3.2 **Improvisação de público**

O diagrama esquemático não representa a improvisação dos usuários. Contudo, os efeitos do ambiente podem ser observados e monitorados pelo sensor de entrada e tratados pela função de revisão de crenças. Desta maneira, caso os usuários venham a provocar alterações no ambiente eles poderão ser observados através destes sensores.

9.3.3 **Improvisação de Resolução de Problemas**

A função filtro controla a adoção de novas intenções com o objetivo de atingir as intenções existentes ou explorar novas oportunidades. Contudo, estas oportunidades estão relacionadas a crenças prévias do agente e para a qual já existem algumas opções disponíveis. O esquema da arquitetura genérica não apresenta como a arquitetura age frente a entradas para as quais não se possui crenças diretamente associadas. Não é mencionado o que ocorre quando as crenças disponíveis não são diretamente mapeadas para a situação do momento, ou seja, a arquitetura não aborda situações que podem acontecer por acaso ou por acidente.

9.4 **Arquitetura BDI para Intenções Conjuntas**

Nesta seção iremos apresentar onde ocorrem implicitamente os tipos de improvisação na arquitetura BDI para intenções conjuntas proposta por Jennings. Detalhes desta arquitetura podem ser obtidos no capítulo 8, seção 8.2.4.

9.4.1 **Improvisação do Texto**

9.4.1.1 *Fase de Preparação*

Na fase de preparação da ação conjunta, o analisador meio-fim, ao decidir qual objetivo deve ser realizado colaborativamente, estabelece uma ação social identificando os possíveis participantes. As intenções conjuntas que são geradas não podem ser executadas diretamente, mas servem como uma descrição que relaciona ações a múltiplos agentes. O relacionamento das ações a serem executadas com os participantes disponíveis é guiado por fatores como estratégia, organização desejada e conhecimento das capacidades dos agentes. Contudo, não é mencionado no trabalho como estes critérios são utilizados para a escolha dos agentes que irão executar as ações. Não parece haver improvisação no sentido da inexistência de alternativas para execução das ações, pois as ações são diretamente relacionadas a um agente. Também não se pode constatar se esta escolha envolve algum aspecto de improvisação no sentido de não ter o agente mais adequado para a ação e como selecionar um outro possível adequado.

9.4.1.2 *Fase de Execução*

Como a execução da ação dentro da ação conjunta é realizada por um agente individual, com base na sua intenção individual, parece existir alguma improvisação, pois é mencionado no artigo que a maneira como o agente irá executar suas ações individuais é deixada em aberto para que o agente escolha a melhor opção de acordo com a situação em que ele se encontra.

9.4.2 **Improvisação da Solução de Problemas**

Na arquitetura proposta por Jennings existe a presença do monitor de eventos, que especifica as razões pelas quais uma ação conjunta deve ser modificada ou finalizada e as ações que devem ser tomadas quando isto acontece. O componente “razões para modificações” define as condições nas quais as ações conjuntas devem ser finalizadas ou novamente analisadas. Depois o agente deve decidir qual ação deve ser realizada para remediar o problema. O modelo das ações associadas descreve atividades apropriadas para as várias classes de eventos. De acordo com Jennings, podem existir vários tipos de eventos que levam um comprometimento ser desfeito.

Novamente existe a idéia de tratar com antecedência a ocorrência de possíveis eventos aos quais o agente apresenta sensibilidade. Contudo, não existe a idéia de eventos que acontecem por acidente ou por acaso e que não foram considerados anteriormente.

9.5 **Arquitetura BDI com Restrições**

Nesta seção iremos apresentar onde estão localizados implicitamente os tipos de improvisação dentro da arquitetura BDI de Chalmers [CHA 2001]. Detalhes desta arquitetura podem ser obtidos no capítulo 8 na seção 8.2.5

9.5.1 **Improvisação do Texto**

9.5.1.1 *Fase de Preparação*

O trabalho se baseia nas idéias de Rao e Georgeff [RAO 95] onde existe um processo de deliberação do próximo curso de ação. O processo de deliberação considera um conjunto de opções e escolhe as opções mais apropriadas. Logo, existe algum tipo de improvisação na montagem do curso de ação.

9.5.1.2 *Fase de Execução*

No início do processo de deliberação o agente constrói um banco de dados de soluções que irá conter todas as soluções possíveis até um determinado nível. O banco de dados de soluções possíveis contém todos os estados possíveis das crenças dos agentes para os próximos n passos especificados. Cada solução candidata do banco de dados possui uma ligação para o estado que o segue. Desta maneira, o agente já possui uma descrição detalhada dos seus possíveis caminhos de execução, o que não permite que o mesmo possua variabilidade e, conseqüentemente, que não improvise. O agente somente escolhe as tarefas que estão mais próximas dele, não existindo improvisação nestas escolhas.

9.5.2 Improvisação de Solução de Problemas

Só consegue solucionar os problemas de tarefas que se sobrepõem. Soluciona isto delegando a tarefa para outro agente. A arquitetura não aborda a questão de ter uma tarefa que não pode ser delegada por falta de capacidade ou habilidade, ou de uma requisição de tarefas para a qual não existe um banco de dados de soluções possíveis diretamente aplicáveis.

9.6 Considerações sobre o Capítulo

Através da análise dos tipos de improvisação presentes nas arquiteturas BDI apresentadas podemos observar a existência da improvisação em alguns processos de planejamento que abordam planos como intenções, ou seja, cursos de ação com alternativas de execução. Entendemos que a visão de planos como intenções possibilita a improvisação e, desta maneira, a improvisação já vem sendo utilizada há bastante tempo de forma intuitiva. Contudo, existem casos onde o planejamento é visto como a idéia de planos como programas e não dão margem para improvisação.

A improvisação que vem sendo utilizada por alguns autores se encontra na construção do curso de ação do agente. Neste sentido, os processos envolvidos com raciocínio meio-fim e deliberação estão relacionados à improvisação do texto, ou seja, a improvisação da construção do curso de ação e os processos relacionados à seleção da ação estão envolvidos na fase de execução da improvisação do texto. Esta improvisação é conhecida no teatro improvisacional como implícita, pois é o tipo de construções que os atores realizam no momento da construção do ato teatral.

Como a improvisação implícita vem sendo utilizada de maneira informal, uma das contribuições da tese é formalizar o seu uso, evidenciando a importância da improvisação como uma característica individual de agentes. Outra contribuição é examinar a questão da improvisação explícita, ou improvisação para solução de problemas, que indica o que fazer quando ocorre algo por acaso ou por acidente. Este tipo de improvisação não foi abordado, até hoje, nos estudos de improvisação computacional.

Na improvisação para solução de problemas, o agente deve utilizar os recursos disponíveis, ou seja, seu conhecimento e habilidades, para gerar uma nova solução para um problema. A geração da nova solução poderia ser feita por analogia, considerando, por exemplo, a proximidade das características do problema atual com algum problema anterior.

No próximo capítulo iremos apresentar um esquema de arquitetura BDI improvisacional, onde incluiremos o módulo de improvisação de solução de problemas dentro da arquitetura BDI e detalharemos como funciona a improvisação de solução de problemas através do uso de analogia.

10 ARQUITETURA BDI IMPROVISACIONAL

No capítulo 6 descrevemos quatro tipos de improvisação e no capítulo 9 analisamos a existência destes tipos de improvisação nas arquiteturas BDI descritas no capítulo 8. A partir desta análise concluímos que as arquiteturas incorporam implicitamente a improvisação de texto, ou seja, a improvisação da construção do curso de ação. Contudo, as arquiteturas não incorporam a improvisação da solução de problemas que ocorrem por acaso e para o qual não se possui nenhum curso de ação pré-estabelecido. Neste capítulo, iremos abordar uma arquitetura de agentes capazes de realizar este segundo tipo de improvisação.

10.1 Diagrama Esquemático da Arquitetura BDI Improvisacional

Tomando como base o diagrama esquemático de Wooldridge apresentado na seção 8.2.3 propomos a extensão da arquitetura BDI tradicional para arquitetura BDI improvisacional. A arquitetura BDI improvisacional possui um novo módulo, o módulo de improvisação de solução de problemas. A figura 10.1 mostra como este módulo pode ser incorporado em uma arquitetura BDI tradicional.

A geração de opções determina as opções disponíveis para o agente com base nas suas crenças, ambiente e intenções atuais. Se a intenção atual do agente é reconsiderar sua intenção para considerar as oportunidades apresentadas pelo ambiente, a função opções é ativada. No caso de algo que aconteceu por acaso ou por acidente, não existe um conjunto de opções de curso de ações disponíveis e conseqüentemente, a geração de opções não consegue gerar um curso de ação. Neste caso, a geração de opções ativa o módulo de improvisação de solução de problemas, que com base em conhecimentos prévios (crenças e intenções) constrói um curso de ação por meio de analogia e transforma este curso de ação em uma nova intenção do agente. A intenção gerada por este módulo segue o padrão das intenções geradas pela função filtro, que vê intenções como curso abstrato de comportamento que deixam a cargo do agente a escolha do que executar no momento adequado.

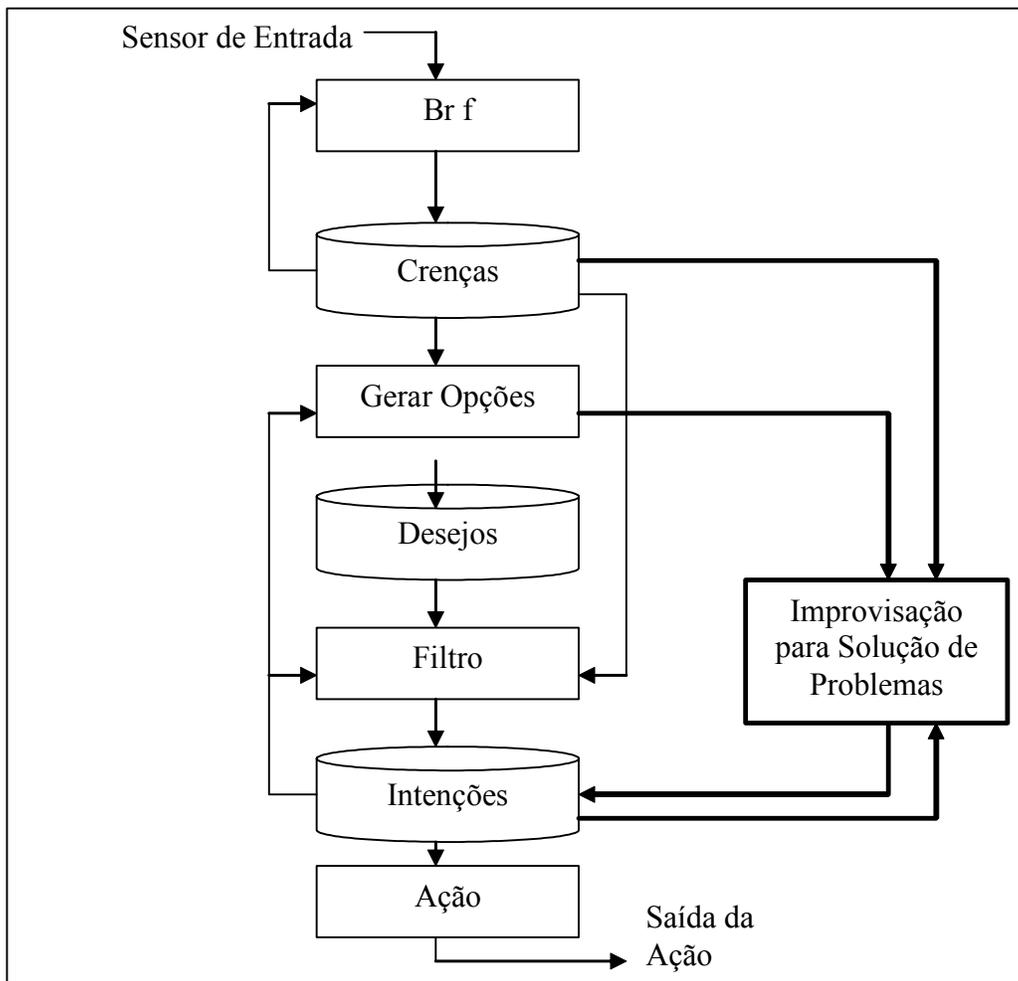


Figura 10.1: Arquitetura BDI Improvisacional como uma extensão do diagrama esquemático de Wooldridge

Seguindo o padrão de apresentação das funcionalidades dos componentes envolvidos no diagrama esquemático da arquitetura BDI de Wooldridge [WOO 99], podemos representar o componente improvisação para solução de problemas através da função improvisação que com base em um conjunto de crenças e intenções gera uma nova intenção para o agente. A função improvisação seria representada como *improvisação*: $\rho(Bel) \times \rho(Int) \rightarrow \rho(Int)$.

O laço de controle tradicional da arquitetura BDI apresentado por Wooldridge [WOO 2000] (figura 8.5) poderia ser estendido como mostra a figura 10.2. O trecho relativo a extensão encontra-se em negrito.

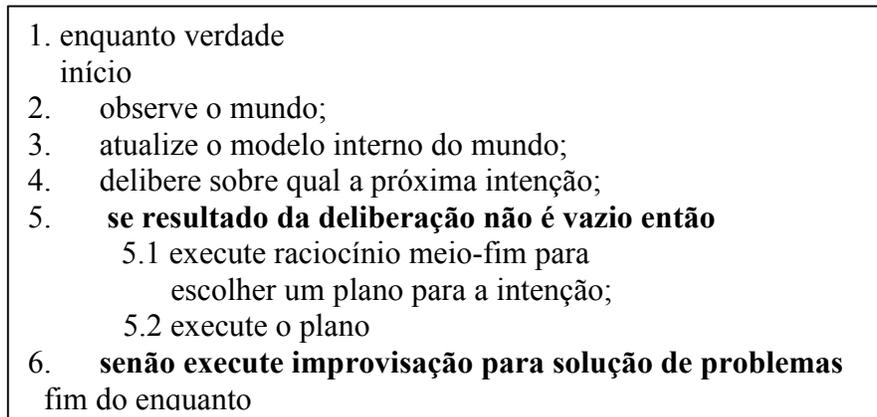


Figura 10.2: Laço de Controle de um Agente BDI Improvisacional

Através desta extensão, a arquitetura BDI improvisacional busca por uma solução usando improvisação como uma analogia a experiências anteriores do agente.

Como mostrado na arquitetura BDI improvisacional (figura 10.1), o módulo de improvisação para solução de problemas executa as funcionalidades da função opções e filtro quando algo acontece por acaso ou por acidente e não existe um conjunto de opções de curso de ações disponíveis para o agente. A função opções é realizada através do submódulo de identificação do tipo de improvisação e a função filtro é realizada através dos submódulos de construção da analogia e transformação da analogia em intenção. Isto indica que a arquitetura para improvisação para solução de problemas também segue um padrão de arquitetura BDI.

10.2 Mapeamento da Terminologia de Crenças, Desejos e Intenções da Arquitetura BDI Tradicional para Arquitetura BDI Improvisacional

Na arquitetura BDI tradicional as crenças, desejos e intenções dos agentes são considerados estados mentais dos agentes. As crenças de um agente representam informações que o agente possui sobre o seu mundo, contendo os estados do mundo e as operações que podem ser realizadas sobre o mundo [WOO 99] [D'IN 97]. As crenças se preocupam com o que o agente acredita e na verdade do que é acreditado [BRA 99a]. Os desejos apresentam um conjunto de opções atuais, representando cursos de ações disponíveis para os agentes [WOO 99]. Os desejos de um agente correspondem intuitivamente às tarefas alocadas para ele. Os agentes devem se fixar em um subconjunto dos desejos e comprometer recursos para realizar estes desejos. Os desejos escolhidos são as intenções do agente [D'IN 97]. Deste modo, as intenções dirigem a ação futura do agente e são controladores de conduta, enquanto os desejos são simplesmente potenciais influenciadores de conduta [BRA 99a].

Na arquitetura BDI improvisacional os estados mentais das crenças, desejos e intenções são manipuladas como objetos. Desta maneira, as crenças representam conhecimento sobre os objetos do ambiente, conhecimento sobre os estados dos objetos e suas operações. Os desejos dos agentes representam a lista de objetos possíveis que o

agente pode considerar para construção de um curso de ação e as intenções são uma descrição abstrata do curso de ação do agente, baseada em uma série de objetos a serem manipulados. Os objetos são descritos em alto nível, desta maneira a intenção pode ser entendida como um plano abstrato que é construído incrementalmente, ao invés de um plano composto de ações primitivas. Maiores detalhes sobre a estrutura da intenção serão apresentados no próximo capítulo.

10.3 Improvisação de Solução de Problemas por Analogia

Nesta seção apresentaremos um embasamento teórico sobre analogia necessário para entendimento do módulo de improvisação para solução de problemas e a nossa proposta para este módulo.

10.3.1 Analogia

De acordo com Salvucci e Anderson [SAL 2001], a analogia tem um papel fundamental na cognição humana. Desde a solução de problemas matemáticos, até a ciência da computação e descoberta criativa, a analogia facilita um melhor entendimento do conhecimento antigo e a formação e inferência do novo conhecimento. A analogia pode ser entendida como o processo de encontrar e usar correspondências entre conceitos.

A capacidade de encontrar e usar correspondências entre conceitos é uma característica interessante para um processo de solução de problemas. De fato, Rich e Knight [RIC 93] salientam que os seres humanos geralmente solucionam problemas fazendo analogias com coisas que eles já sabem fazer. Desta maneira, a dificuldade estaria em determinar que coisas são semelhantes e que coisas não são. Esta é a tarefa do raciocínio por analogia.

Antes de discutir o uso da analogia no raciocínio, é importante analisar o conceito de analogia. De acordo com o dicionário Aurélio [FER 99], analogia significa: ponto de semelhança entre coisas diferentes; semelhança, similitude; identidade de relações entre os termos de dois ou mais pares e semelhança de função entre dois elementos, dentro de suas respectivas totalidades. Sowa e Majumbar [SOW 2003] apresentam conceitos mais especializados de analogia como: similaridade em alguns aspectos de coisas que de outra maneira não são similares; comparação que determina o grau de similaridade e uma inferência baseada na semelhança ou correspondência.

Através dos conceitos de analogia concluímos, assim como Rich e Knight [RIC 93], que a analogia é uma abordagem interessante para solução de problemas, em especial problemas que acontecem por acaso e para o qual o agente não possui possíveis cursos de ação. Este tipo de problema ativa no agente improvisacional sua capacidade de improvisação explícita, e por este motivo entendemos solução de problemas como improvisação de solução de problemas, sendo esta improvisação alcançada por analogia, onde o agente usa conhecimento existente para interpretar novas situações.

10.3.1.1 Definição de Analogia

O estudo de analogia em IA vem sendo realizado desde 1960 [SOW 2003] e freqüentemente os pesquisadores utilizam definições de analogia especializadas para uma

situação particular. Contudo, todos concordam que a analogia fornece um mecanismo básico para efetivamente conectar um raciocínio passado com uma experiência presente.

Na analogia os aspectos conhecidos de um novo caso são comparados com aspectos correspondentes de casos antigos. O caso antigo que possui a melhor combinação de aspectos correspondentes pode ser assumido como a melhor fonte de evidência para estimar os aspectos não conhecidos do novo caso. Quanto maior a concordância entre as alternativas dos aspectos não conhecidos, mais forte a evidência para uma conclusão do novo caso [SOW 2003].

De acordo com Hall [HAL 89] a analogia consiste de um conjunto de relações em comum entre as partes do problema. Analogia é descrita como um mapeamento entre elementos de um domínio fonte e um domínio alvo e se torna particularmente interessante quando o raciocínio é familiar com a fonte e pode mapear elementos ou relações familiares da fonte em elementos ou relações não familiares (não conhecidos) em um alvo.

A analogia tem sido utilizada em várias áreas da IA como dedução automática, solução de problemas e planejamento, processamento da linguagem natural e aprendizagem de máquina. Hall, em [HAL 89], realiza uma revisão de trabalhos em analogia computacional nestas áreas e propõe uma estrutura para raciocínio analógico que deve incluir os seguintes componentes:

1. reconhecimento de uma fonte análoga candidata, dada uma descrição do alvo;
2. elaboração de um mapeamento analógico entre os domínios fonte e alvo, possivelmente incluindo um conjunto de inferências analógicas;
3. avaliação do mapeamento e inferências em algum contexto de uso, incluindo justificativa, reparo ou extensão do mapeamento;
4. consolidação da saída da analogia, a fim de que seus resultados possam ser re-instanciados para outros contextos.

Usando esta estrutura Hall analisa diversos trabalhos em analogia computacional. Segundo Salvucci e Anderson [SAL 2001] a peça central da analogia é o processo de mapeamento analógico. De fato, ao analisar os trabalhos apresentados em Hall constatamos que cada pesquisador possui uma versão para o seu mapeamento de analogia. Como neste trabalho estamos abordando a questão de solução de problemas e planejamento, iremos descrever brevemente alguns dos trabalhos apresentados por Hall nesta área.

10.3.1.1.1 McDermott [MCD 78][MCD 79]: Aprendendo a usar analogias

McDermott descreve ANA, um sistema de produção implementado que usa analogias quando está resolvendo problemas em um ambiente simulado. No sistema ANA, o raciocínio analógico assimila novos problemas para um determinado conjunto de métodos enquanto um mecanismo de aprendizado simples acomoda os resultados da solução do problema para aperfeiçoar a performance. A assimilação cobre o reconhecimento, elaboração e avaliação de analogias, enquanto a acomodação corresponde à consolidação. A estrutura para solução de problemas é um sistema de produção tradicional, onde um ciclo de reconhecimento/ação aplica produções cujos elementos condição combinam com os elementos da memória de trabalho. As produções de propósito especial suportam cada processo de analogia pela manipulação de um mapeamento entre os elementos da memória

de trabalho do problema alvo e os elementos selecionados de um método fonte mais adequado. O sucesso ou falha de um método aplicado é armazenado por produções construídas que transpassam a analogia quando o problema alvo é visto novamente.

ANA executa em um ambiente simulado de uma loja de pintura, onde o conjunto de objetos reconhecidos podem ser encontrados, movidos, e pintados com tinta ou água. As entradas incluem a localização dos objetos, seis métodos para realizar objetivos específicos (por exemplo, produções como pinte uma mesa em L32 de vermelho) e suporte a fontes de conhecimento codificadas como regras de produção. Estas regras definem tipos de hierarquia de objetos e ações, reconhecem analogias entre problemas e métodos, e gerenciam o mapeamento de elementos para analogias, caminhos de recuperação de erros na aplicação de métodos e armazenam a experiência. Quando as fontes de conhecimento do ANA são insuficientes, o usuário é solicitado a diagnosticar e corrigir erros na aplicação do método. Como saída o sistema resolve tarefas por analogia e adquire produções para resolver estas tarefas mais rapidamente em uma segunda apresentação.

Ao ser solicitada a realização de uma tarefa para a qual não existem métodos diretamente aplicáveis, ANA tenta reconhecer uma analogia com um destes métodos. McDermott usa dois mecanismos para produzir uma analogia entre a nova tarefa e o método análogo. Primeiro, ele indexa cada método com uma “produção de descrição do método” que irá disparar sempre que tipos similares de ações ou objetos ocorrem na tarefa alvo. Segundo, ele adiciona um processo de extração que re-expressa ações e objetos no problema alvo, generalizando-os ou especializando-os em relação aos tipos de hierarquias conhecidas. Estas manipulações garantem que os índices de métodos são eventualmente disparados. Por exemplo, um método *painting a table in L32* pode ser reconhecido como método fonte para *washing a safe in L12* pela generalização e especialização da ação alvo (por exemplo, *wash/spray/paint*) e o objeto alvo (por exemplo *safe/thing/table*).

Quando um índice de método dispara, ele *asserts* os elementos possíveis com mapeamento em problemas alvos correspondentes. Os elementos alvo são dados em uma declaração do problema, detectados pelo ambiente ou estipulados como elementos esperados. Logo, cada elemento *asserted* pela produção da descrição do método é mapeado em um elemento alvo, conhecido ou esperado para existir. Na analogia entre *washing* e *painting* do exemplo anterior, o mapeamento analógico inicial inclui:

Paint → *wash* ,

table → *safe*,

L32 → *L12* ,

red → *clean-surface* .

10.3.1.1.2 Carbonell [CAR 83] [CAR 86]: resolução de problemas reconstitutivos

Carbonell propõe dois métodos distintos para analogia: transformacional e derivacional. O método de analogia transformacional transforma um caminho solução de um problema anterior (fonte) em um caminho solução para o problema alvo. Um resolvidor de problemas tradicional de análise meio-fim é estendido em um espaço de problema de transformação de analogia (*T-space*). *T-space* é um espaço de problemas de segunda ordem na qual os estados encapsulam uma seqüência de chamadas de operadores no espaço original. Cada operador *T-space* (*T-operator*) modifica a seqüência para dar uma nova

seqüência de solução. A idéia é que a solução de problemas analógicos pode ocorrer no *T-space* como uma rotina ocorre na solução de problemas em um espaço de problemas tradicional. Quando as condições dos *T-operators* são satisfeitas, a aplicação do operador fornece um nova seqüência e a seqüência resultante eventualmente satisfaz a especificação do objetivo alvo. A figura 10.3 ilustra o método de analogia transformacional.

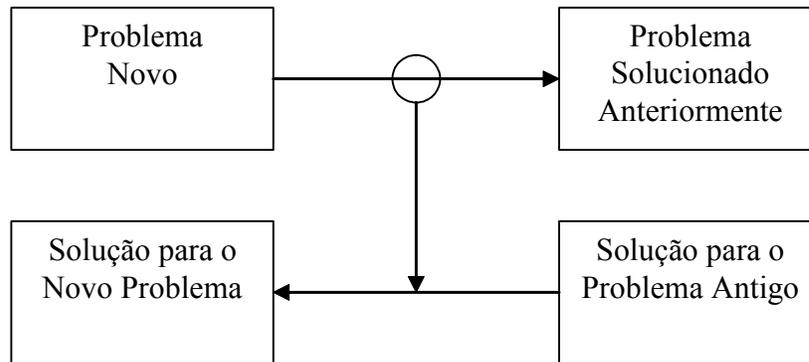


Figura 10.3: Analogia Transformacional (retirada de [RIC 93])

O sistema ARIES é uma implementação inicial do método transformacional. Dado um problema alvo descrito no espaço de estados original (estado inicial, estado objetivo e restrições) ARIES seleciona uma solução fonte com descrições de estados e restrições similares. A seqüência de passos na prova fonte é o estado inicial no *T-space*, e uma seqüência que prova o teorema alvo é o estado objetivo. A analogia é elaborada no *T-space* por uma busca meio-fim por uma seqüência de transformações que produzem uma seqüência de solução alvo.

O reconhecimento ocorre com base em um mapeamento parcial entre a descrição do problema alvo e os episódios da solução fonte, indexados em uma memória hierarquicamente organizada. A solução fonte mais promissora é incrementalmente transformada em uma seqüência de operadores que fornece uma solução alvo correta. Carbonell reporta que a implementação inicial de ARIES foi efetiva quando testada em vários domínios, incluindo problemas de álgebra e tarefas de rotas de planejamento, mas foi incapaz de tirar vantagem de informação mais abstrata de planejamento que poderia ser armazenada com uma seqüência solução (por exemplo, razões para decisões de solução de problemas ou falhas).

O método de analogia transformacional não analisa como o problema foi resolvido, ele só analisa a solução final. O método de analogia derivacional estende a abordagem transformacional para incluir passos derivacionais completos para problemas anteriormente solucionados. A derivação inclui subobjetivos; escolhas alternativas de operadores em cada passo com razões para decisão entre elas; nodos iniciais e terminais para caminhos sem sucesso com razões para a falha; interdependência entre decisões sucessivas; ponteiros para estruturas periféricas de conhecimento usadas durante a solução de problemas e razões e/ou suposições que justificam o sucesso ou insucesso das tentativas de solução de problemas. Carbonell argumenta que estas informações são necessárias para transformar com sucesso uma solução fonte para uma solução para o novo problema. A figura 10.4 ilustra o método de analogia derivacional.

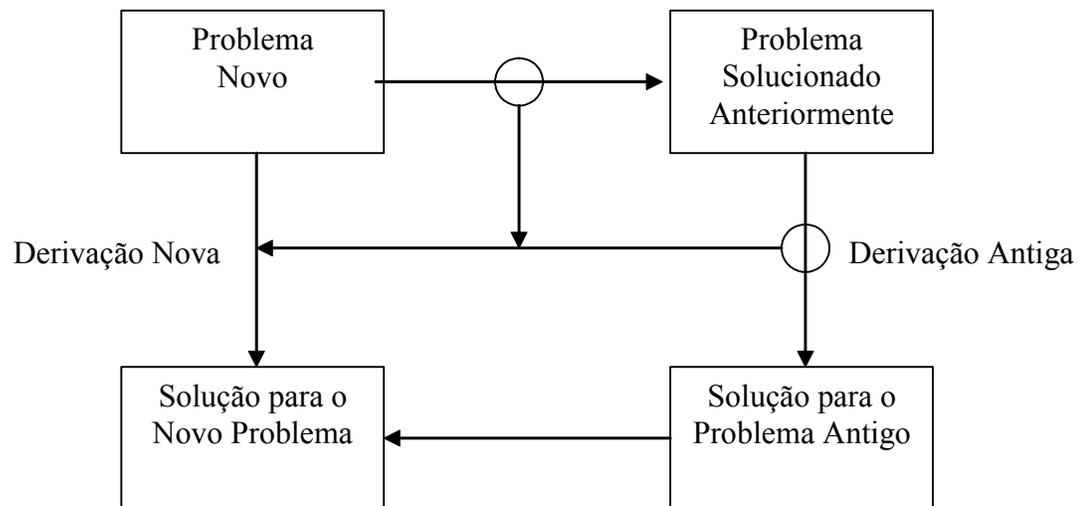


Figura 10.4: Analogia Derivacional (retirada de [RIC 93])

O método de analogia derivacional estende a analogia transformacional em dois aspectos. Primeiro, as justificativas guiam o processo de reconstrução, possibilitando ao mecanismo de analogia avaliar não somente o caminho de solução atual (por exemplo, uma série de operadores), mas também um registro de porque passos particulares foram realizados e tiveram sucesso ou falharam. Isto oferece uma vantagem sobre um método relativamente menos informado de transformações baseadas na diferença. Segundo, a abordagem derivacional integra o reconhecimento de analogias diretamente no contexto de solução de problemas que estão sendo solucionados.

Os trabalhos apresentados enfocam o uso de uma memória anterior e enfatizam o planejamento como busca em um espaço de estados para encontrar a melhor analogia. Isto ocorre porque está sendo utilizado o método tradicional de planejamento para solução de problemas. Entendemos que esta abordagem não é adequada para nossos fins, pois a construção de uma analogia demandaria muito tempo desta maneira. Optamos por utilizar outro método ao invés das técnicas tradicionais para solução de problemas, um método que incorpore a construção de analogias por similaridade. O método escolhido foi a analogia por similaridade proposta por Russell [RUS 86][RUS88][DAV 87].

10.3.1.2 Analogia por Similaridade

Na ausência de informação específica relevante, a suposição tradicional no estudo da analogia tem sido de que o análogo mais similar fornece, com maior probabilidade, uma solução correta. Russell [RUS 86] mostra como uma análise estatística pode ser executada a fim de fornecer a probabilidade de que uma determinada fonte irá produzir uma analogia com sucesso, usando somente a suposição de que existem algumas características relevantes nas descrições fonte e alvo. De acordo com Russell, a utilidade da analogia por similaridade parece se basear em algumas suposições muito fundamentais sobre a natureza das nossas representações.

O raciocínio analógico é normalmente definido como o argumento de similaridades conhecidas entre duas coisas para a existência de futuras similaridades. Formalmente, a analogia é definida como qualquer inferência que segue o esquema:

anal

$$P(S,A), P(T, A), Q(S, B) \rightarrow Q(T, B)$$

onde T é o alvo sobre o qual se quer conhecer algum fato Q (consulta); S é a fonte, o objeto a partir do qual irá se obter a informação para satisfazer Q por analogia; P representa as similaridades conhecidas dadas pelos valores de atributos compartilhados A. P e Q podem ser fórmulas arbitrárias de cálculo de predicados e A e B podem ser tuplas arbitrárias de objetos.

Um número incontável de inferências tem esta forma, mas esta forma não distingue entre inferências analógicas boas ou ruins. A abordagem tradicional para decidir se uma analogia é razoável, tem sido dizer que cada similaridade observada contribui com alguma evidência extra para a conclusão. Isto leva naturalmente a suposição de que a fonte de analogia mais apropriada é aquela que tem a maior similaridade para com o alvo. Contudo, na literatura filosófica sobre analogia, vários autores têm observado a inadequação da similaridade contada por argumentos como a base para a analogia, particularmente quando muitas analogias são convincentes.

De acordo com Russell [RUS 88], Hesse observou a relevância das similaridades conhecidas para as similaridades inferidas. A teoria das determinações [RUS 86] [DAV 87] fornece uma definição para relevância. Esta teoria suporta que dado que as similaridades conhecidas são (parcialmente) relevantes para as similaridades inferidas, a inferência analógica tem a garantia de ser (parcialmente) justificada. O fato de que P é relevante para Q é codificado como uma determinação, escrita como $P(\underline{x}, \underline{y}) \phi Q(\underline{x}, \underline{y})$ e definido como:

$$\forall \underline{wxyz} P(\underline{w}, \underline{y}) \wedge P(\underline{x}, \underline{y}) \wedge Q(\underline{w}, \underline{z}) \Rightarrow Q(\underline{x}, \underline{z})$$

Quando o raciocínio tem este tipo de informação disponível, a atenção pode ser direcionada para aquelas similaridades que são relevantes para o problema que se tem, e a justificativa da conclusão é lógica por natureza e o grau total de similaridade não tem mais lugar no processo.

Russell [RUS 88] propõe que pelo menos um aspecto de um sistema de raciocínio analógico consista de um processo probabilístico para determinações parciais. O argumento crucial para a analogia baseada em determinação é que a determinação representa aquela classe de regularidades cuja extrapolação toma a forma de um raciocínio analógico; sem a habilidade de usar e detectar determinações, um sistema é simplesmente empobrecido de seu poder inferencial.

Antes de se construir uma teoria de determinação é necessário se resolver os seguintes problemas:

1. uma maneira razoável deve ser encontrada para circunscrever as descrições fonte e alvo. Sem isto, o conjunto de fatos a serem comparados é essencialmente sem limites;
2. uma medida de similaridade deve ser definida de maneira a ser (tanto quanto possível) independente da maneira como a fonte e alvo são representadas;
3. deve-se identificar as suposições necessárias para relacionar a medida de similaridade a uma probabilidade desejável.

A essência da abordagem é mostrar que a analogia a uma fonte mais similar pode ser justificada em termos da ausência de qualquer determinação usável. Isto significa que em situações onde o sistema é extremamente ignorante, existirão muitas determinações (fatores causais) que ele desconhece. Alguns fatos poderiam ser relevantes para a pesquisa mesmo que não se tenha razão direta para se acreditar nisto. Neste caso, uma similaridade mais ampla serve para aumentar o grau de semelhança com que tais fatores serão levados em consideração, aumentando a chance de que características relevantes irão ser incluídas nos aspectos comuns.

Se uma fonte combina com um alvo em todas as características relevantes, uma analogia desta fonte é assumida ser correta. Para uma consulta ter uma solução, é necessário que todas as características relevantes para uma consulta apareçam em algum lugar na descrição do alvo a ser combinado com a fonte. Quando isto não acontece é necessário se ter uma conclusão probabilística, que será apresentada a seguir.

Um modelo por analogia simplificado em um banco de dados é o seguinte: existe um alvo T descrito por m pares de atributos-valores, para o qual se quer encontrar o valor de outro atributo Q. Existem várias fontes S_1, \dots, S_n (análogas) que têm valores para o atributo Q desejável bem como para os m atributos conhecidos para o alvo. A similaridade s é definida como o número de valores de atributos que combinam para um determinado alvo e fonte e a diferença é definida por $d = m - s$. Assumindo que existem r atributos relevantes para descobrir o valor de Q, $p(d, r)$ é definida como sendo a probabilidade que a fonte S, diferindo do seu alvo em d atributos, se combina com os r atributos relevantes. A suposição de que não existe informação relevante significa que todos os atributos são igualmente importantes para serem relevantes. Pode-se calcular $p(d, r)$ usando um argumento combinatório simples [RUS 88]:

Seja N_m o número de escolhas de quais atributos são relevantes tais que S combina com T em tais atributos. Seja N o número total de escolhas de quais atributos são relevantes.

$$p(d, r) = N_m / N$$

$$= \frac{\binom{m-d}{r}}{\binom{m}{r}} \quad (r \geq 1)$$

Através desta probabilidade a analogia mais similar é garantida ser a analogia mais desejável. Quanto maior o número de atributos relevantes é necessário uma combinação maior de atributos para garantir que as similaridades relevantes estão realmente presentes.

A suposição de um valor fixo para as características relevantes não parece ser realística. A suposição mais geral que se pode fazer é que r siga uma distribuição de probabilidade $q_Q(r)$ que depende do tipo de consulta Q . Embora isto introduza um grau extra de liberdade dentro da teoria, os autores descobriram que os resultados são quase independentes do que se assume sobre q . Nesta nova versão se calcula a probabilidade de sucesso da analogia como uma função somente da diferença d entre fonte e alvo, usando a fórmula anterior para $p(d, r)$. Desta maneira, a nova versão seria:

$$p(d) = \sum_{r=0}^m q(r)p(d,r)$$

Para qualquer suposição razoável sobre a forma de $q(r)$, a variação de $p(d)$ com d permanece aproximadamente da mesma forma.

Pode-se fazer com que o modelo simples analisado acima possa ser aplicável a qualquer tarefa analógica simplesmente permitindo que os atributos e valores sejam fórmulas e termos arbitrários de cálculo de predicados.

Russell [RUS 86][RUS 88] acredita que, quando sistemas inteligentes que englobam teorias completas de racionalidade limitada são construídos, a habilidade de executar raciocínio analógico usando determinações e similaridades será essencial a fim de permitir ao sistema usar sua experiência de maneira proveitosa. Analogia por similaridade também parece ser bem aplicada para a tarefa de produzir respostas plausíveis e rápidas para problemas. Desta maneira, a analogia por similaridade parece preencher os requisitos necessários para a execução da improvisação para solução de problemas.

10.3.1.3 Analogia e Raciocínio Baseado em Casos

Ao estudarmos analogia nos deparamos com a área de raciocínio baseado em casos. Segundo Aamodt e Plaza [AAM 94], raciocínio baseado em casos (CBR – *case-base reasoning*) é capaz de utilizar conhecimento específico de situações de problemas (casos) concretos e já vividos para resolver novos problemas. Um novo problema é resolvido encontrando um caso passado similar, e reusando este caso na nova situação do problema. Desta maneira, o raciocínio baseado em casos resolve um problema lembrando uma situação similar anterior e reusando informação e conhecimento daquela situação. Através deste conceito podemos constatar uma similaridade entre analogia e raciocínio baseado em casos, pois estudos de solução de problemas por analogia também mostram o uso freqüente de experiência passada na solução de problemas novos e diferentes.

De fato, o raciocínio baseado em casos e analogia são algumas vezes utilizados como sinônimos[AAM 94]. Contudo, Aamodt e Plaza [AAM 94] acreditam que os métodos de analogia resolvem novos problemas baseados em casos passados para domínios diferentes, enquanto métodos baseados em casos tipicamente enfocam a indexação e combinação de estratégias para casos em um domínio único. Embora os autores apresentem esta diferenciação iremos considerar neste trabalho que analogia e raciocínio baseado em casos são sinônimos, visto que as etapas executadas pela analogia e raciocínio baseado em casos são muito semelhantes, para não dizer as mesmas. A tabela 10.1 apresenta a descrição destas etapas.

Tabela 10.1: Comparação entre as Etapas da Analogia e do Raciocínio Baseado em Casos

Etapas do Raciocínio por Analogia	Etapas do Raciocínio Baseado em Casos
1. reconhecimento de uma fonte análoga candidata, dada uma descrição do alvo;	1. recuperar o caso ou casos mais similares;
2. elaboração de um mapeamento analógico entre os domínios fonte e alvo, possivelmente incluindo um conjunto de inferências analógicas;	2. reusar a informação e conhecimento do caso para resolver um problema;
3. avaliação do mapeamento e inferências em algum contexto de uso, incluindo justificativa, reparo ou extensão do mapeamento e;	3. revisar a solução proposta;
4. consolidação da saída da analogia, a fim de que seus resultados possam ser re-instanciados para outros contextos.	4. reter as partes desta experiência que podem ser úteis para a solução de problemas no futuro.

10.3.2 Arquitetura e Funcionamento do Módulo de Improvisação de Solução de Problemas

O módulo de improvisação de solução de problemas tem por objetivo produzir uma intenção para um agente quando ocorre algo inesperado, para o qual não existe um curso de ação diretamente aplicável. Quando uma situação acontece no ambiente, ela é percebida pelo sensor de entrada do agente que ativa uma função de revisão de crenças. Após as crenças terem sido atualizadas, o agente ativa a função de geração de opções e a função filtro que decide se a intenção deve ou não ser reconsiderada. Caso a intenção deva ser reconsiderada, o fluxo volta para a função opções que deve gerar alternativas possíveis para um curso de ação. No caso de uma situação inesperada que aconteceu por acaso ou acidente, o agente não consegue gerar alternativas e, a fim de não falhar, ele ativa o módulo de improvisação de solução de problemas.

O módulo recebe como entrada uma descrição do objeto que ocasionou o problema, e as crenças atuais do agente e gera uma intenção, um curso de ação apropriado, para a situação. A geração da intenção é realizada através da construção de uma analogia a uma intenção existente, que soluciona um objeto de características similares ao atual.

10.3.2.1 Descrição das Fontes e Alvo

Como apresentado em Russell [RUS 88], antes de trabalhar com a teoria da determinação utilizada no raciocínio por similaridade é necessário se encontrar uma maneira de descrever as características das fontes e do alvo. Entendemos que tanto as fontes quanto o alvo devem ser descritos através de um modelo genérico de objetos. Propomos um modelo em que os objetos podem ser de dois tipos: metas e meios. Cada um dos tipos de objetos é descrito através de grupos de atributos.

Os objetos classificados como do tipo metas estão relacionados ao objetivo do agente para um determinado curso de ação. Os objetos classificados como do tipo meio estão relacionados ao uso dos mesmos para viabilizar um determinado curso de ação. Deste modo, os grupos de atributos do tipo meta são objetivo, efeito e características particulares, e o grupo de atributos do tipo meio são uso possível e características particulares (figura 10.5). O grupo de uso possível possui uma lista de usos de um objeto. O grupo de características particulares possui um conjunto de atributos e valores que especificam as características do objeto. Os valores dos grupos de atributos (objetivo, efeito, listas de usos possíveis dos objetos e características particulares) são dependentes do domínio que se deseja modelar. Ao se aplicar um domínio particular sobre o modelo genérico, se obtém um modelo concreto de objetos.

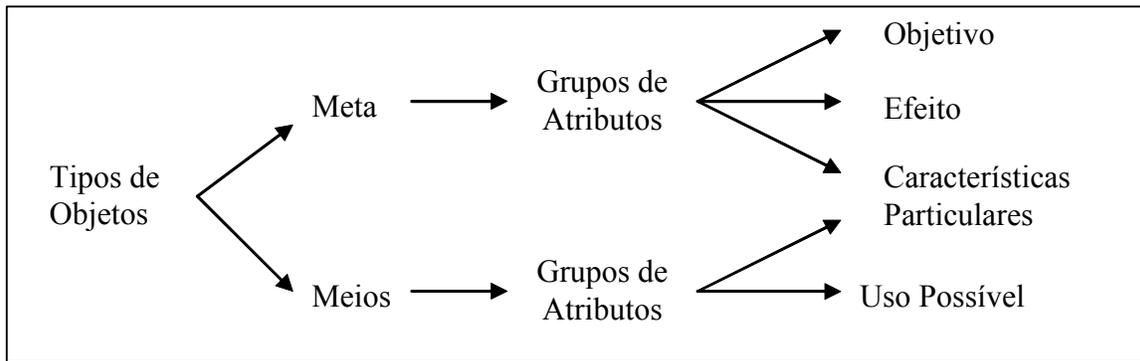


Figura 10.5: Modelo genérico de objetos

Um objeto meta pode estar relacionado a um objeto meio através de suas características particulares. Por exemplo, um objeto meta pode ter como característica particular o nome de um objeto meio que faz parte do seu curso de ação.

10.3.2.2 Tipos de Improvisação de Solução de Problemas

Como mencionado anteriormente, o módulo de improvisação de solução de problemas irá ser acionado quando não existir um curso de ação diretamente aplicável à situação que o agente se encontra. Este fato pode ser ocasionado devido a dois problemas: o agente não conhece um meio de alcançar um objetivo da maneira como ele foi descrito ou o agente desconhece o objetivo da maneira como ele foi descrito. Isto leva a existência de dois tipos de improvisação:

- no uso de um objeto como um meio improvisado para alcançar o objetivo que se está trabalhando;
- no uso de um objeto como um objetivo improvisado alternativo ao objetivo que estava sendo trabalhado inicialmente.

No primeiro caso temos improvisação em nível de maneiras de se alcançar um objetivo e no segundo caso improvisação em nível de um novo objetivo, já conhecido por experiências anteriores e que pode ser aplicado através de similaridade ao objetivo atual. Na próxima seção mostraremos como o módulo irá abordar estes diferentes tipos de improvisação.

10.3.2.3 Arquitetura do Módulo de Improvisação de Solução de Problemas

A figura 10.6 apresenta a arquitetura do módulo de improvisação de solução de problemas com sua entrada e saída. O módulo é composto de três submódulos: identificação do tipo de improvisação, construção da analogia e transformação da analogia em intenção. A descrição do objeto problema segue o mesmo padrão de um objeto do tipo meio, composto de usos possíveis e características particulares.

10.3.2.3.1 Submódulo Identificação do Tipo de Improvisação

O submódulo identificação do tipo de improvisação recebe a descrição do objeto problema e após executar envia para o submódulo construção da analogia uma identificação do tipo de improvisação a ser executado. Para identificar se deve ser realizada uma improvisação de meio ou uma improvisação de objetivo, o submódulo procura por objetos do tipo meta, no conjunto de crenças, que tenham um objetivo compatível com os usos possíveis do objeto problema. Se for encontrado um ou mais objetos compatíveis, o tipo de improvisação a ser realizado é de meio. Neste momento, o submódulo procura por objetos do tipo meio, no conjunto de crenças, que tenham usos possíveis e características particulares compatíveis com a descrição do objeto problema, gerando uma lista de objetos meio análogos possíveis.

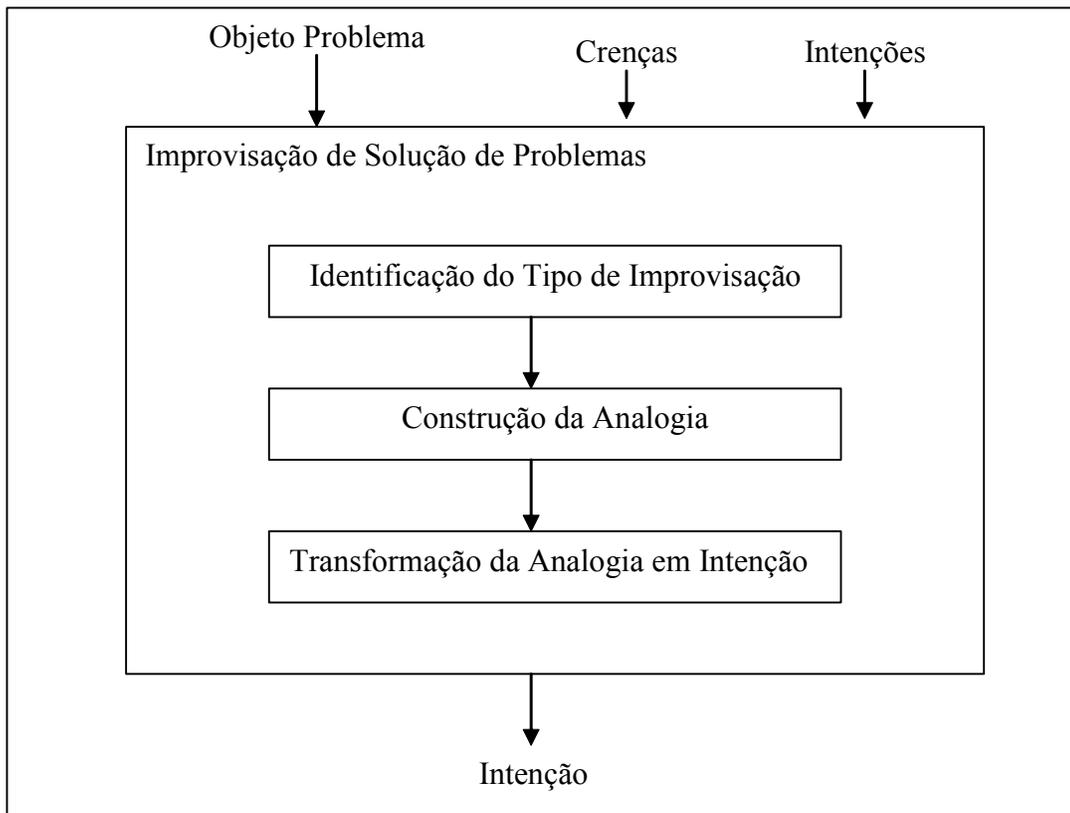


Figura 10.6: Arquitetura do Módulo de Improvisação para Solução de Problemas

Se não for encontrado nenhum objeto do tipo meta compatível com o objeto problema, o tipo de improvisação a ser realizado é de objetivo. Neste caso, o submódulo deve procurar por objetos do tipo meta, no conjunto de crenças, que possuam efeito compatível com os usos possíveis da descrição do objeto problema, gerando uma lista de objetos meta análogos possíveis. Caso a lista de objetos seja vazia, ou seja, não foi encontrado nenhum objeto fonte que tivesse efeito compatível com o uso possível descrito no objeto problema, ocorre uma improvisação de objetivo sem efeito conhecido, onde o submódulo procura por objetos meio que possuam características particulares semelhantes às características particulares e usos possíveis do objeto problema, gerando uma lista de objetos meta-meio possíveis.

A figura 10.7 descreve na forma de um algoritmo de alto nível o processo envolvido neste submódulo.

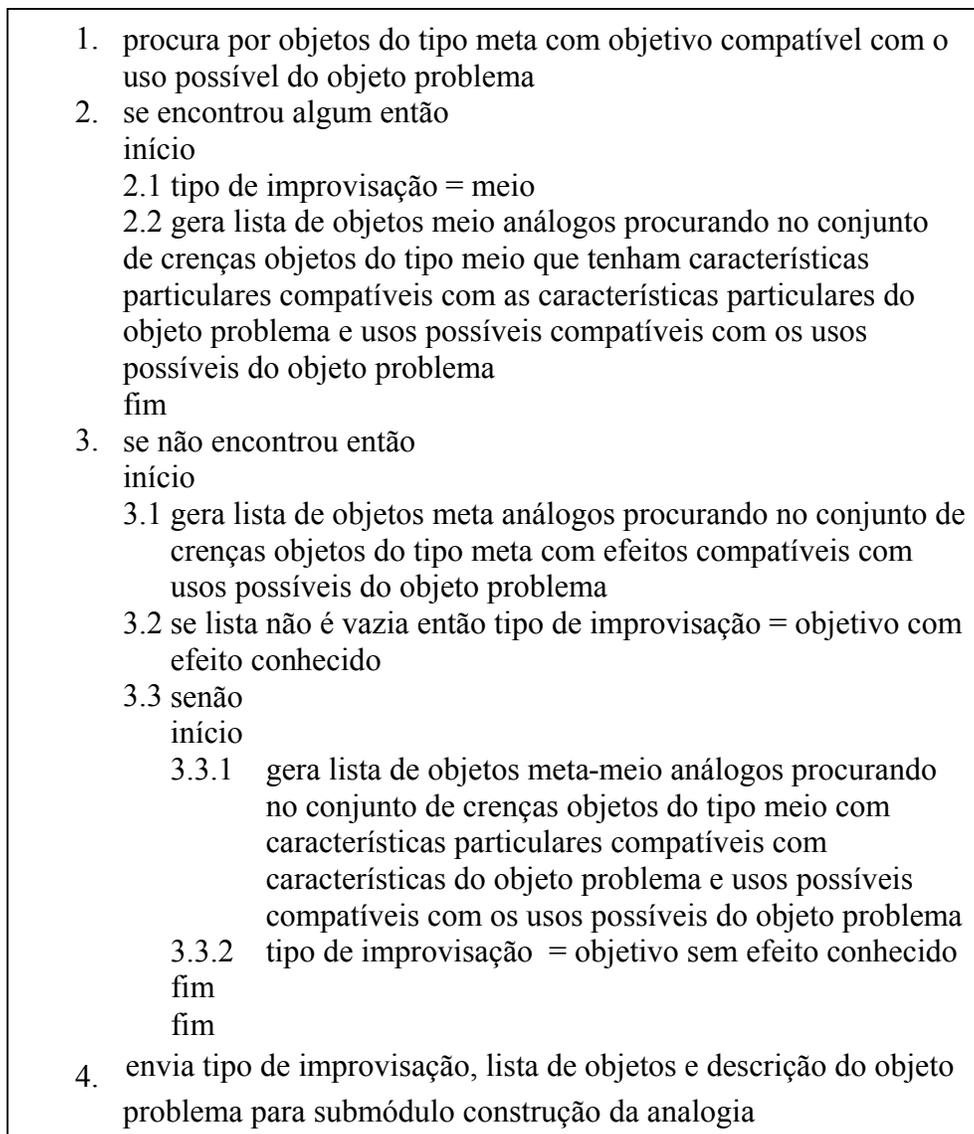


Figura 10.7: Algoritmo de alto nível para submódulo identificação do tipo de improvisação

Desta maneira, o submódulo implementa improvisação de meio e dois tipos de improvisação de objetivos. O primeiro tipo, chamado de improvisação de objetivo com efeito conhecido, possui relacionamento entre os efeitos de um objeto com os usos possíveis de outro. O segundo tipo, chamado de improvisação de objetivo sem efeito conhecido, não possui relacionamento entre os efeitos e usos possíveis, mas possui algum relacionamento entre as características particulares do objeto fonte com as características e usos possíveis do objeto problema. As improvisações de meio e de objetivo com efeito conhecido, geram uma analogia “forte”, enquanto a improvisação de objetivo sem efeito conhecido gera uma analogia “fraca”. A analogia é dita “forte” para os casos onde existe um relacionamento dos usos possíveis com os objetivos dos objetos meta ou existe um relacionamento entre os usos possíveis com os efeitos dos objetos meta. A analogia é dita “fraca” para o caso onde não existe relacionamento entre usos possíveis e objetivos, de maneira direta ou indireta, mas existe somente algum relacionamento entre as características particulares. Estes dois tipos de analogia fazem com que o agente sempre consiga construir uma analogia a um problema que aconteceu por acaso ou por acidente, mesmo que esta analogia não seja a mais adequada possível.

O submódulo identificação do tipo de improvisação gera duas saídas para o submódulo construção da analogia. A primeira é o tipo de improvisação (meio, objetivo com efeito conhecido e objetivo sem efeito conhecido) e a segunda é uma lista contendo as possíveis fontes análogas. Esta lista é gerada durante o processo de descoberta do tipo de improvisação a ser realizado. Além destas saídas, o submódulo passa a descrição do objeto problema para o próximo submódulo.

10.3.2.3.2 Submódulo Construção da Analogia

Este submódulo recebe a descrição do objeto problema, a indicação do tipo de improvisação de solução de problemas que deve ser realizada e uma lista das possíveis fontes análogas. Independente do tipo de improvisação a ser executado, iremos utilizar as idéias apresentadas por Russell para seleção da fonte análoga mais provável de ter sucesso, dada uma descrição alvo. Para tanto será definida a similaridade s , a diferença d , os atributos relevantes r , e a probabilidade $p(d,r)$ para cada uma das possíveis fontes análogas geradas pelo módulo de identificação do tipo de improvisação.

Se a solução de problemas a ser realizada é improvisação de meio, a similaridade s será calculada através do número de valores de atributos que combinam para um determinado alvo e fonte, considerando os atributos do grupo de características particulares e usos possíveis. No caso da improvisação de objetivo com efeito conhecido, a similaridade será calculada considerando os atributos do grupo efeito. Isto ocorre, na medida em que, ao mudar de objetivo, as características particulares podem não ser necessariamente compatíveis, mas o uso possível do objetivo original para improvisação com efeito conhecido é compatível. No caso da improvisação de meio, o objetivo não muda, no sentido de que é um objetivo conhecido do agente, mas como o objetivo pode não ser o objetivo da intenção atual, utiliza os usos possíveis para procurar por similaridade dentro dos objetos do tipo meio.

No caso da improvisação de objetivo sem efeito conhecido, a similaridade s é calculada considerando o grupo de características particulares e usos possíveis, embora não se possua relação entre o uso possível do objeto problema e os efeitos dos objetos meta.

Esta analogia é considerada como “fraca” devido a esta falta de relação. Isto significa que esta analogia não é ideal, mas evita do agente falhar ao tentar solucionar um problema que aconteceu ao acaso ou por acidente.

Após calcular a similaridade da fonte análoga, deve-se calcular a diferença entre os m atributos que descrevem o alvo e a similaridade s , como $d = m - s$. Assume-se a existência de r atributos relevantes, como um argumento de entrada e se calcula $p(d, r)$ como sendo a probabilidade que a fonte S, diferindo do seu alvo em d atributos, se combina com os r atributos relevantes. A suposição de que não existe informação relevante significa que todos os atributos são igualmente importantes para serem relevantes.

Seja N_m o número de escolhas de quais atributos são relevantes tais que S combina com T em tais atributos. Seja N o número total de escolhas de quais atributos são relevantes, a probabilidade pode ser calculada através de uma combinação simples:

$$p(d, r) = N_m / N$$

$$= \frac{\binom{m-d}{r}}{\binom{m}{r}} \quad (r \geq 1)$$

Esta probabilidade irá garantir que a analogia mais similar é a analogia mais desejável. Este cálculo de probabilidade é aplicado considerando um número fixo para as características relevantes. Pode-se fazer com que r siga uma distribuição de probabilidade $q_Q(r)$ que depende do tipo de consulta Q . Nesta nova versão se calcula a probabilidade de sucesso da analogia como uma função somente da diferença d entre fonte e alvo, usando a fórmula anterior para $p(d, r)$.

$$p(d) = \sum_{r=0}^m q(r)p(d, r)$$

Embora isto introduza um grau extra de liberdade dentro da teoria, Russell e Davies [RUS 88] [DAV 87] descobriram que os resultados são quase independentes do que se assume sobre q .

Nesta tese está sendo considerada a existência de um conjunto fixo de atributos para descoberta da probabilidade da melhor fonte análoga. Desta maneira, a função de probabilidade utilizada será a da combinação simples $p(d, r) = N_m / N$.

O cálculo da similaridade s , atributos do alvo m e atributos relevantes r , são calculados para a improvisação de meio e improvisação de objetivo sem efeito conhecido considerando os grupos de características particulares e de usos possíveis. Já para realizar o cálculo dos mesmos na improvisação de objetivos com efeito conhecido, é considerado somente o grupo de atributos efeito.

O submódulo deve aplicar a fórmula de probabilidade ($p(d, r) = N_m / N$) para cada um dos possíveis objetos análogos da lista fornecida pelo submódulo de identificação do tipo de improvisação. O objeto análogo escolhido será aquele que tiver a maior probabilidade de ser semelhante ao alvo. Ao final da sua execução, o submódulo envia para o submódulo transformação da analogia em intenção o tipo de improvisação realizado e a

fonte análoga selecionada. A figura 10.8 mostra o algoritmo de alto nível do módulo de construção da analogia.

1. se tipo improvisação igual a meio então
 - 1.1 enquanto lista de objetos meio análogos é diferente de vazio
 - início
 - 1.1.1 calcula similaridade s considerando características particulares e usos possíveis
 - 1.1.2 calcula diferença d entre m atributos que descrevem o objeto problema e a similaridade s
 - 1.1.3 atribui a r o número de atributos relevantes
 - 1.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 1.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 - fim_enquanto
2. se tipo improvisação igual a objetivo com efeito conhecido então
 - 2.1 enquanto lista de objetos meta análogos é diferente de vazio
 - início
 - 2.1.1 calcula similaridade s considerando efeitos
 - 2.1.2 calcula diferença d entre m atributos que descrevem o objeto problema e a similaridade s
 - 2.1.3 atribui a r o número de atributos relevantes
 - 2.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 2.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 - fim_enquanto
3. se tipo improvisação igual a objetivo sem efeito conhecido então
 - 3.1 enquanto lista de objetos meta-meio análogos é diferente de vazio
 - início
 - 3.1.1 calcula similaridade s considerando características particulares e usos possíveis
 - 3.1.2 calcula diferença d entre m atributos que descrevem o objeto problema e a similaridade s
 - 3.1.3 atribui a r o número de atributos relevantes
 - 3.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 3.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 - fim_enquanto
4. seleciona da lista de analogia o objeto que possui maior probabilidade de similaridade
5. envia tipo de improvisação e objeto selecionado para submódulo da transformação da analogia em intenção

Figura 10.8: Algoritmo de alto nível do submódulo de construção da analogia

10.3.2.3.3 Submódulo Transformação da Analogia em Intenção

O submódulo transformação da analogia em intenção recebe o tipo de improvisação e o objeto fonte que é a analogia mais apropriada para a descrição do objeto problema e o transforma em uma descrição de intenção. Se o tipo de improvisação é de meio ou objetivo sem efeito conhecido, a intenção a ser gerada é uma modificação da intenção atual para conter o objeto selecionado como mais similar ao objeto problema. Se o tipo de improvisação é de objetivo com efeito conhecido, é gerada uma nova intenção a partir do objeto meta selecionado. Como mencionado anteriormente, um objeto meta possui relacionamentos com objetos meios através de suas características particulares. O algoritmo de alto nível do submódulo da transformação da analogia em intenção pode ser visualizado na figura 10.9.

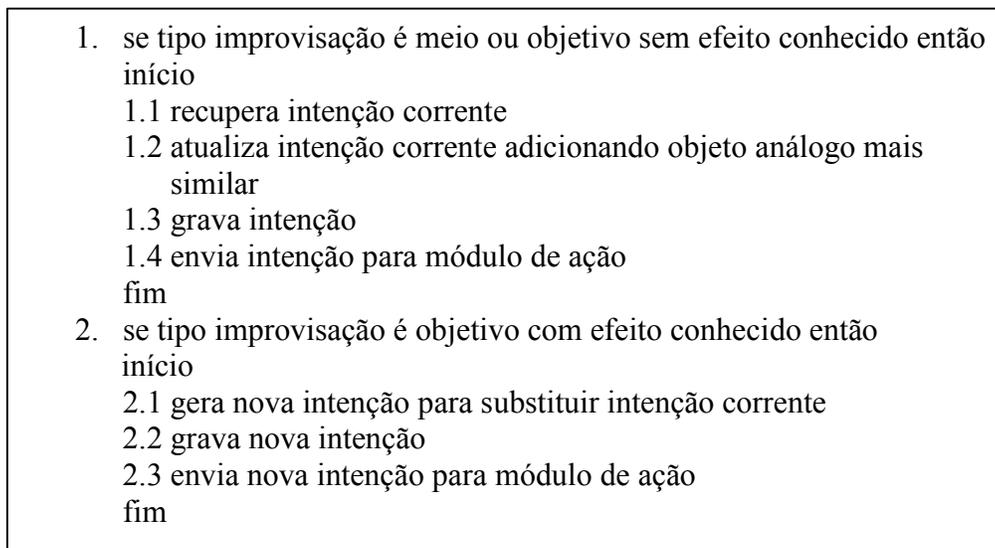


Figura 10.9: Algoritmo de alto nível para submódulo de transformação da analogia em intenção

A intenção a ser modifica ou gerada neste submódulo segue o mesmo padrão das intenções anteriores, ela descreve um curso de comportamento abstrato para o agente não especificando a ação no seu nível primitivo, de ação como execução. Desta maneira, a intenção gerada pela improvisação de solução de problemas segue os mesmos moldes das intenções geradas por uma arquitetura BDI tradicional. A representação da intenção irá determinar a maneira como ela será construída. Deste modo, cada arquitetura para um domínio específico irá possuir um mecanismo próprio de transformação da analogia gerada em uma intenção do agente.

10.4 Definição de Agentes Improvisacionais como Agentes Deliberativos

Propomos que agentes improvisacionais sejam definidos como agentes deliberativos que possuem capacidade de gerar e executar comportamentos de forma improvisada e de resolver problemas utilizando conceitos de improvisação. Quando a arquitetura de um

agente improvisacional é baseada em uma arquitetura BDI improvisacional, além das funções de opções e filtro que geram alternativas para os cursos de ação do agente, ela deve possuir um módulo de improvisação para solução de problemas, responsável por solucionar problemas que ocorrem por acaso ou acidente e para o qual o agente não possui uma informação diretamente aplicável. A improvisação neste módulo deve ser realizada com base na analogia por similaridade.

10.5 Considerações sobre o Capítulo

O módulo de improvisação para solução de problemas tem como objetivo agir quando as funções tradicionais do modelo BDI não conseguem gerar opções para um curso de ação que solucione uma situação que ocorreu por acaso ou acidente. Neste sentido, o módulo de improvisação para solução de problemas pode ser entendido como responsável pela reconsideração de intenções quando o método tradicional realizado pela função filtro e opções não é possível.

A improvisação para solução de problemas foi desenvolvida utilizando a idéia de analogia por similaridade proposta por Russell [RUS 88]. Das quatro etapas envolvidas no processo de analogia, a saber, identificação das fontes análogas, mapeamento entre as fontes análogas e alvo, avaliação da analogia e consolidação da analogia, somente não realizamos a terceira. A primeira foi realizada através do submódulo identificação do tipo de improvisação, a segunda foi realizada através do submódulo construção da analogia e a última foi realizada através do submódulo de transformação da analogia em intenção.

Para realizarmos estas etapas e utilizarmos a teoria de determinação da analogia por similaridade descrevemos um modelo de objetos genérico e os tipos de improvisação que o módulo poderia realizar. Identificamos dois tipos de improvisação: de meio e de objetivo. Este último é executado através de duas maneiras diferentes, uma onde os efeitos do objeto fonte são compatíveis com o uso possível do objeto problema e outra onde os efeitos não são compatíveis, mas existem características particulares compatíveis. O primeiro caso foi chamado de improvisação de objetivo com efeito conhecido e o segundo caso de improvisação de objetivo com efeito não conhecido.

A simulação do funcionamento do módulo através de um modelo de objetos concretos será realizada nos dois próximos capítulos, onde será apresentada uma arquitetura BDI improvisacional para guias virtuais e a aplicação desta arquitetura em um estudo de caso no museu virtual SAGRES.

Através do módulo de improvisação para solução de problemas, apresentamos uma abordagem para a definição de improvisação explícita como algo realizado sem preparo prévio e fazendo uso de recursos disponíveis. O agente utiliza os recursos disponíveis, ou seja, seu conhecimento e habilidades, descritos como um conjunto de objetos, e a partir daí gera uma nova solução. Isto é realizado através da analogia por similaridade entre o objeto problema e os objetos conhecidos pelo agente.

11 APLICAÇÃO DA ARQUITETURA BDI IMPROVISACIONAL PARA UMA ARQUITETURA DE GUIAS VIRTUAIS

Nesta seção iremos apresentar a aplicação da arquitetura BDI Improvisacional em uma arquitetura para guias virtuais que se baseia nos fundamentos teóricos do Teatro Improvisacional. Guias virtuais são agentes responsáveis por preparar e apresentar conteúdos para usuários. Desta maneira, a arquitetura pode ser composta por guias virtuais que assumem papéis de agentes atores e diretores ou diretores-atores, sendo todos eles agentes improvisacionais. A composição da arquitetura irá definir o tipo de coordenação improvisacional que ocorre na sociedade de agentes, sendo esta centralizada ou descentralizada, como discutido no capítulo 7.

Quem é responsável por realizar a coordenação improvisacional é o diretor. Neste sentido é importante considerar as características de um diretor improvisacional, descritas no capítulo 5, para construção de uma arquitetura que possibilite este tipo de coordenação.

De acordo com Morrison [MOR 74], Black [BLA 91] e Berger [BER 98] uma das funções mais importantes de um diretor improvisacional é a interpretação de uma peça e o treinamento e coordenação de atores. Este treinamento e coordenação é passada para os atores através de *scripts*. Para construir os *scripts* o diretor deve interagir com o escritor da peça, obtendo informações sobre a peça, interpretando-as e inserindo sua visão criativa particular. Além disso, Spolin [SPO 99] salienta que o diretor deve auxiliar os atores a resolver problemas que podem surgir no decorrer de uma atuação teatral e realizar uma avaliação da performance dos atores bem como da sua performance.

A arquitetura proposta aborda os processos envolvidos nas atividades de construção de *scripts* abstratos de comportamento (SAC), ou intenção, e solução de resolução de problemas, ou improvisação para solução de problemas. Cabe salientar que estas atividades podem ser diretamente mapeadas dentro da arquitetura BDI Improvisacional descrita no capítulo 10. A arquitetura de guias virtuais pode ser visualizada na figura 11.1.

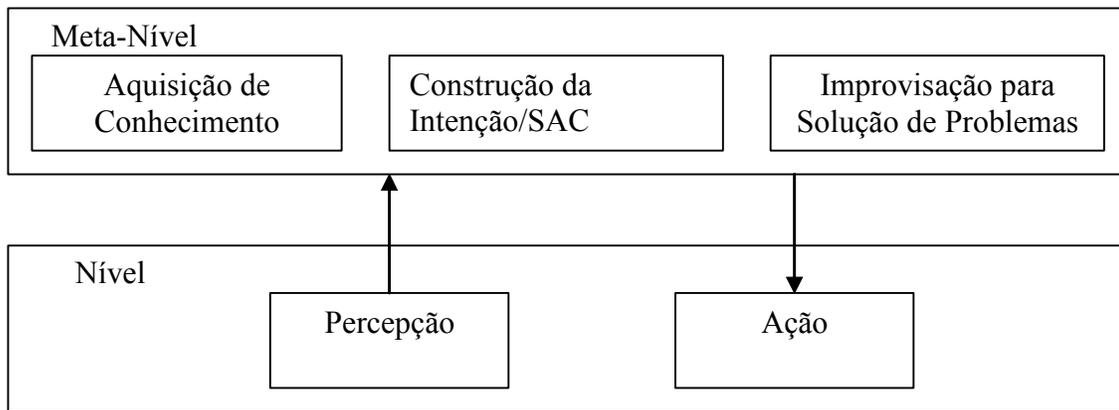


Figura 11.1: Arquitetura Genérica de Guias Virtuais usando Arquitetura BDI Improvisacional

A arquitetura está dividida em nível e meta-nível (figura 11.1) [MAE 88]. No nível encontram-se os módulos de atuação e percepção e no meta-nível encontram-se os módulos que requerem o conhecimento de como o agente executa os processos de aquisição de conhecimento, construção de intenção/SAC e improvisação para solução de problemas.

Mapeando a arquitetura genérica de guias virtuais (figura 11.1) para o diagrama esquemático da arquitetura BDI improvisacional (figura 9.1) observamos que a aquisição de conhecimento representa a função de revisão de crenças da arquitetura BDI improvisacional. A construção de intenção/SAC engloba as funções de opções e filtro. A improvisação para solução de problemas representa o módulo de improvisação para solução de problemas. A percepção representa o sensor de entrada e a ação representa a função de ação na arquitetura BDI improvisacional.

Como apresentado no capítulo 9 as arquiteturas BDI tradicionais examinadas podiam executar somente a improvisação de texto que está relacionada à construção do curso de ação do agente. A nossa arquitetura pode executar tanto a improvisação de texto, através do seu módulo de construção da intenção/SAC, quanto a improvisação de solução de problemas, executada pelo módulo com o mesmo nome. Cabe salientar que o módulo de construção da intenção/SAC também poderia realizar a improvisação de personalidade do ator, caso ele possuísse uma representação gráfica e uma personalidade.

A arquitetura apresentada na figura 11.1 irá se adaptar ao papel que o agente está executando em um determinado momento. O agente pode ser um diretor, um diretor-ator ou um ator. Por exemplo, no papel de diretor ou diretor-ator é de responsabilidade do agente adquirir conhecimento para construção do *script*. No caso do agente no papel de ator isto não é necessário, pois ele irá receber um *script* abstrato de comportamento e será responsável por executá-lo. Os agentes operam sobre as mesmas estruturas, mas o nível de abstração está relacionado ao papel que o agente executa em um determinado momento. Desta maneira, quando o agente está no papel de diretor ele executa improvisação no nível de direção, quando está no papel de ator ele executa improvisação no nível de atuação e quando está no papel de diretor-ator, executa improvisação tanto de direção quanto de atuação. Através da apresentação da interação entre os módulos e da sua descrição serão

esclarecidas a existência e utilidade dos mesmos em cada um dos papéis que os agentes podem atuar.

11.1 Tipos de Coordenação Improvisacional Presentes na Arquitetura

A arquitetura incorpora dois tipos de coordenação improvisacional apresentados no capítulo 7, a coordenação centralizada e a descentralizada. Na coordenação centralizada existe um agente diretor e vários agentes atores. O diretor é responsável por coordenar seus atores através da construção de intenções como *scripts* abstratos de comportamento e da solução de problemas que podem ocorrer durante a execução destas intenções. Os agentes atores são responsáveis por improvisar comportamentos com base nos *scripts* abstratos recebidos.

Na coordenação descentralizada existem diretores-atores, ou seja, o agente executa tanto as funcionalidades do diretor quanto as funcionalidades do ator. Neste caso, o agente possui autonomia suficiente para construir sua própria intenção e executá-la. Ele irá atuar em dois papéis distintos em determinados momentos. No início da execução ele deve operar como diretor para construir sua intenção como um *script* abstrato de comportamento, depois passa a atuar como ator e executar seus comportamentos de maneira improvisada. Ele somente irá retomar o papel de diretor quando ocorrer algum problema durante a execução ou quando tiver finalizado a sua intenção.

11.2 Interação entre os Componentes da Arquitetura

Os componentes da arquitetura de guias virtuais irão interagir de diferentes maneiras dependendo do tipo de papel que o agente está executando. Neste sentido, serão apresentadas as interações que ocorrem quando o agente está no seu papel de diretor, ator e diretor-ator. No papel de diretor-ator o agente realiza as interações do diretor e as interações do ator dependendo do papel que estiver executando em um determinado momento.

11.2.1 Interação entre os Componentes da Arquitetura para o Papel de Diretor

O diretor inicia o seu processo interagindo com o autor para aquisição de conhecimento sobre uma determinada apresentação. O autor se comunica com o diretor através do módulo de percepção. Ao finalizar a aquisição de informação, o módulo de aquisição de conhecimento interage com o módulo de construção da intenção/SAC. Após o módulo de construção da intenção/SAC executar, o mesmo interage com o módulo da ação para enviar uma intenção a um determinado ator. Neste momento, o módulo de percepção fica sensível a qualquer requisição de um agente ator. Ao receber uma percepção, o módulo identifica o tipo da mesma, e encaminha para o módulo da construção da intenção no caso de uma requisição de fim de execução de intenção, ou para o módulo de improvisação de solução de problema no caso de uma falha.

Ao receber uma percepção de falha, o módulo improvisação para solução de problemas realiza improvisação como analogia por similaridade para solucionar o problema identificado. Se a solução do problema requer a atualização da intenção atual, o módulo faz isto e envia para o módulo de ação a atualização da intenção corrente para que seja enviada ao agente ator que gerou a falha. Caso a solução do problema exija a construção de uma

nova intenção, o módulo de improvisação de solução de problemas solicita ao módulo de construção da intenção/SAC, a construção da mesma. Após construir uma nova intenção, a mesma é enviada para o módulo de improvisação de solução de problemas que ativa o módulo de ação para interagir com o agente ator.

A figura 11.2 ilustra a interação descrita nos parágrafos acima.

11.2.2 Interação entre os Componentes da Arquitetura para o Papel de Ator

O ator inicia o seu processo recebendo do diretor, através do seu módulo de percepção, uma intenção como um *script* abstrato de comportamento (SAC). Após o módulo de construção da intenção do ator escolher o que deve ser executado, ele envia a solicitação de execução para o módulo de ação que interage com o ambiente externo. Enquanto isto o módulo de percepção fica aguardando alguma requisição do ambiente externo. Quando chega uma requisição, o módulo de percepção envia a mesma para o módulo de construção da intenção/SAC. Se a percepção indicar a execução de alguma atividade da intenção como SAC, o módulo se comunica com o módulo da intenção e o ciclo se repete. Caso a requisição provoque uma falha, o módulo de construção da intenção/SAC ativa o módulo de improvisação para solução de problemas, que faz uma requisição de solução de problema para o diretor através do módulo de ação.

A figura 11.3 ilustra a interação descrita nos parágrafos acima.

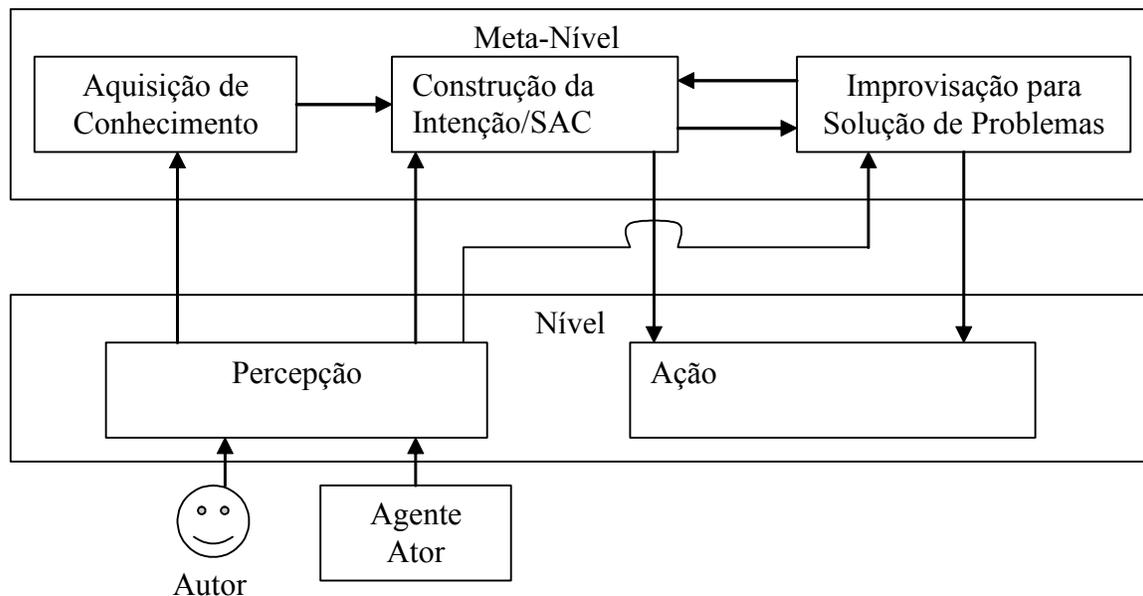


Figura 11.2: Interação entre os componentes da arquitetura para o papel de diretor

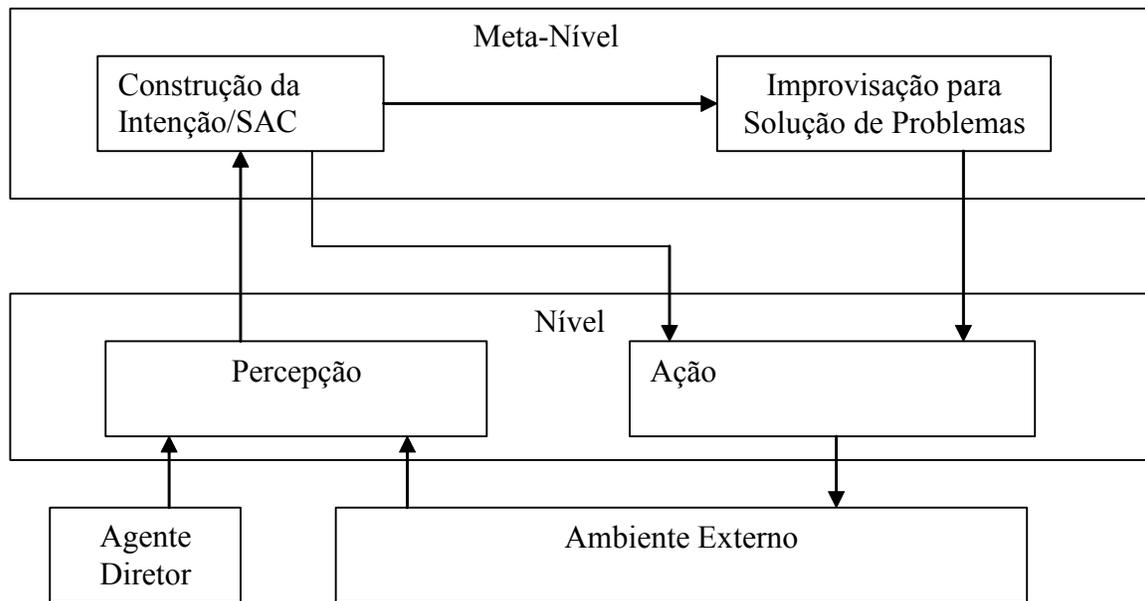


Figura 11.3: Interação entre os componentes da arquitetura para o papel de ator

11.3 Módulo de Aquisição de Conhecimento

O módulo de aquisição de conhecimento, como o próprio nome sugere tem como objetivo adquirir conhecimento para futura construção dos *scripts* abstratos de comportamento, ou intenções, que serão passadas para os agentes. O agente somente poderá utilizar este módulo quando ele estiver executando o papel de diretor ou diretor-ator, pois de acordo com os pressupostos do teatro virtual somente o diretor pode interagir com o escritor ou roteirista para receber informações da peça e posteriormente comunicar aos atores através de *scripts*. Desta maneira, este módulo envolve uma interação direta entre o escritor e o agente no seu papel de diretor.

O diretor deve solicitar ao escritor informações sobre o curso de ação na forma de uma seqüência, mais ou menos abstrata, de atividades que o agente deve executar. Primeiro o diretor deve solicitar informações sobre o curso de ação. Depois ele deve solicitar informações sobre as atividades que formam o curso de ação, e por último deve solicitar a estrutura do conteúdo. As atividades e os conteúdos devem ser descritos de maneira a possibilitar o uso destas informações no momento da construção de uma improvisação por analogia para solução de problemas que podem ocorrer durante a execução. Deste modo, ao descrever as atividades e os conteúdos, devem ser informados os seus usos possíveis e suas características particulares.

A troca de informações entre o diretor e o escritor ocorre através de mensagens, como mostra a figura 11.4.

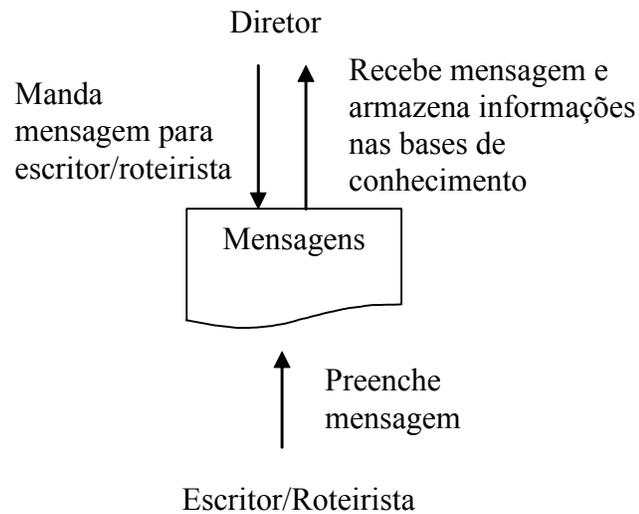


Figura 11.4: Interação entre diretor e escritor na aquisição de conhecimento

O tipo de mensagem que está sendo trocada implica na atualização de uma base de conhecimento específica. Se a mensagem é para aquisição de informações sobre o curso de ação, é atualizada a base de conhecimento de cursos de ação. Se a mensagem é para aquisição de informações sobre a atividade, a base de conhecimento atualizada é a de atividades. Se a mensagem é para aquisição de informações sobre o conteúdo, a base atualizada é a de conteúdo.

O funcionamento do módulo de aquisição de conhecimento é descrito através de um algoritmo de alto nível, escrito em português, e mostrado na figura 11.5.

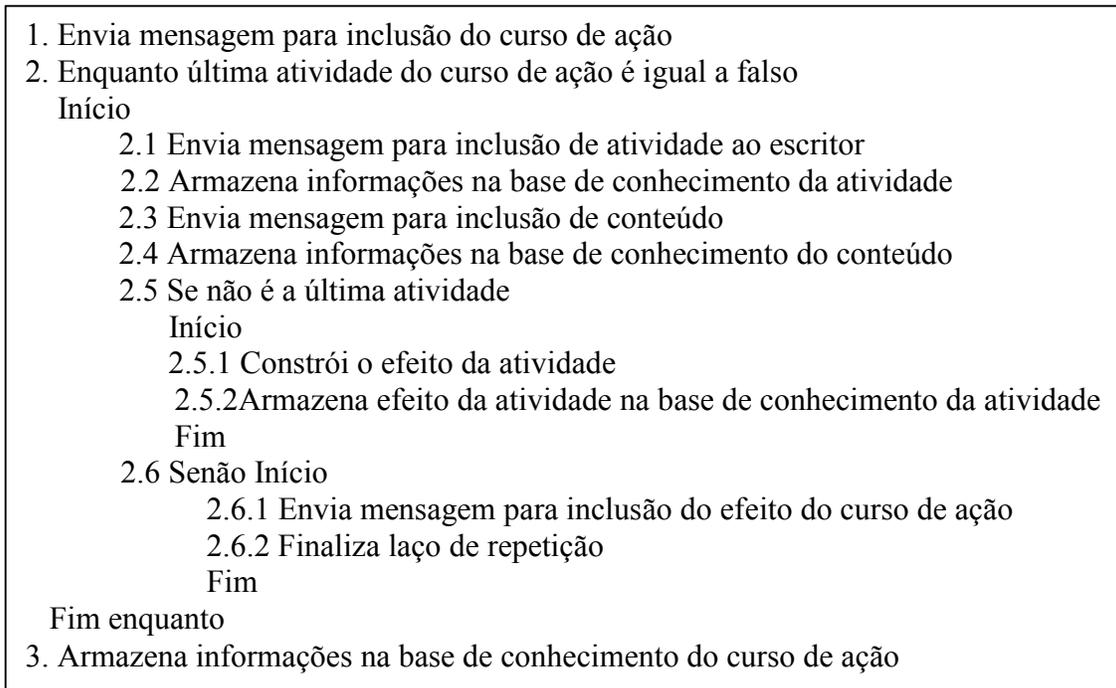


Figura 11.5: Algoritmo de alto nível do módulo de aquisição de conhecimento

Este algoritmo também pode ser descrito através do protocolo apresentado na figura 11.6.

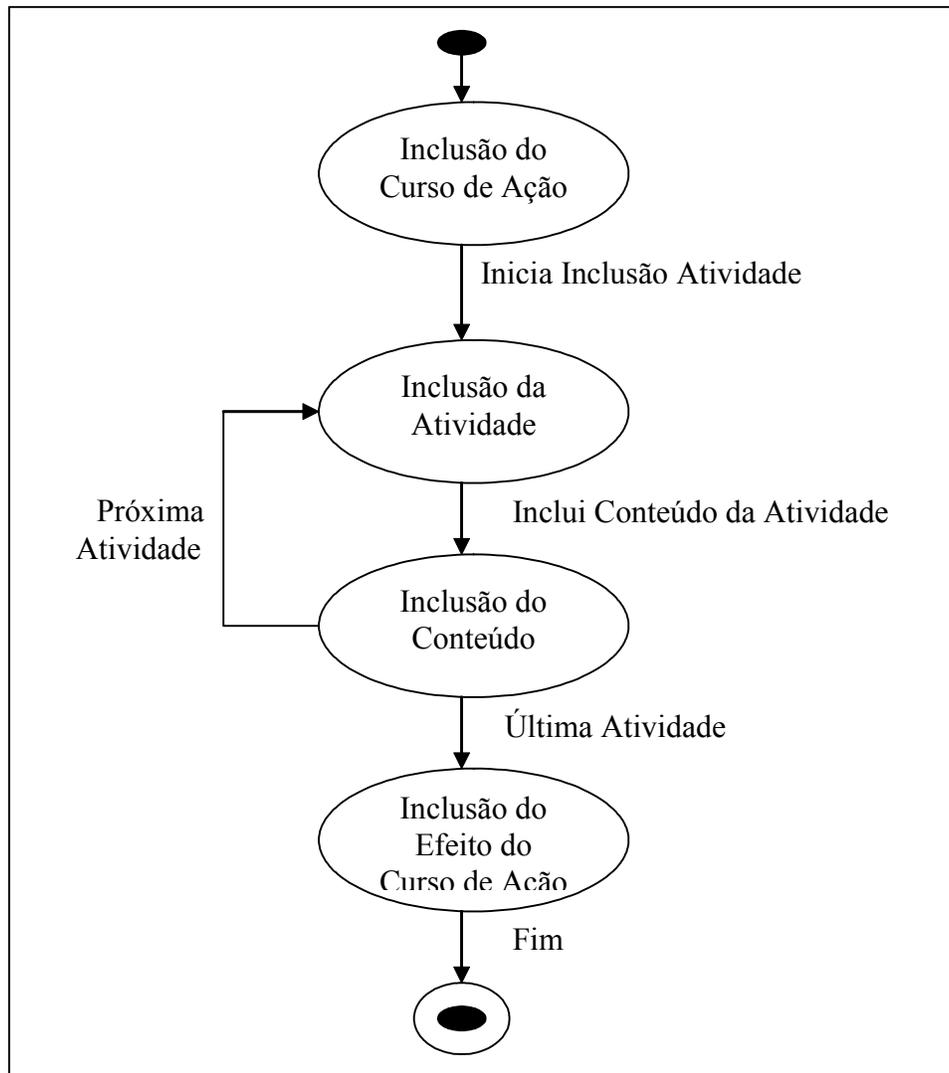


Figura 11.6: Protocolo para aquisição de conhecimento

A base de conhecimento do curso de ação irá armazenar as informações de acordo com a estrutura apresentada na tabela 11.1:

Tabela 11.1: Estrutura da base de informação do curso de ação

Código	Objetivo	Efeito	Atividade Inicial
--------	----------	--------	-------------------

Onde, o código representa o identificador do curso de ação, o objetivo informa o objetivo do curso de ação, o efeito informa o que ocorre ao se executar o curso de ação e a atividade inicial aponta para a primeira atividade do curso de ação. O código e atividade inicial são as características particulares do objeto curso de ação.

A base de conhecimento da atividade irá armazenar informações de acordo com a estrutura apresentada na tabela 11.2.

Tabela 11.2: Estrutura da base de informação da atividade

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas

A estrutura contém os usos possíveis e as características particulares. Os grupos de características particulares definidos para atividade de um guia virtual são: nome, precedência, efeito e físicas. Os usos possíveis indicam as possíveis utilidades desta atividade, o nome indica o nome da atividade, a precedência indica a atividade que deve ter sido realizada anteriormente para que esta possa ser executada, o efeito indica o que ocorre quando esta atividade é realizada e físicas indica quais movimentos estão relacionados a esta atividade. O agente pode ter um certo número de características físicas diferentes, o que contribui para a sua variabilidade de comportamento.

A base de conhecimento do conteúdo irá armazenar informações de acordo com a estrutura apresentada na tabela 11.3.

Tabela 11.3: Estrutura da base de informação do conteúdo

Usos Possíveis	Características particulares			
	Tipo	Área	Subárea	Textos

Do mesmo modo que a estrutura da atividade, a estrutura do conteúdo possui os usos possíveis e as características particulares. As características particulares definidas para um conteúdo são: tipo, área, subárea e textos. O tipo de conteúdo indica para o agente se o conteúdo é específico de uma área que o agente conhece ou se é um conteúdo de conhecimento geral; a área indica o enfoque de concentração do conteúdo; a subárea indica qual tópico da área é abordado e os textos indicam o conhecimento propriamente dito que o agente possui sobre um conteúdo. O campo texto pode possuir um certo número de opções de textos diferentes para um determinado conteúdo. Isto possibilita ao agente uma variação do seu comportamento verbal, o que contribui para a sua credibilidade.

Exemplos de tipos de conhecimento, áreas e subáreas podem ser visualizados na figura 11.7. A classificação do conhecimento como geral ou específico irá depender do tipo de domínio que o agente deve apresentar. No exemplo abaixo, o agente considera como conteúdos específicos conhecimentos sobre matemática e história e como conteúdos gerais suas informações pessoais e dados sobre cinema.

Cabe salientar que as características particulares da atividade e do conteúdo podem ser estendidas para representar outros grupos de características. Esta expansão está relacionada ao domínio que se quer modelar. As características particulares descritas nas tabelas 11.2 e 11.3 são consideradas as características básicas.

Tipo	Área	Subárea
Geral	Pessoal	Identificação Gostos Parentes
Geral	Cinema	Aventura Drama Comédia
Específico	Matemática	Figuras Funções Operações
Específico	História	Dinossauros Egípcios Gregos

Figura 11.7: Exemplos de tipos, áreas e subáreas de conhecimento

As estruturas armazenadas no módulo de aquisição de conhecimento possibilitam que a improvisação de texto ocorra de duas maneiras. Primeiro, através da seqüência de atividades que pode ser informada de uma maneira abstrata, não especificando completamente a ordem em que as tarefas devem ser executadas. Segundo, pelas possibilidades de variação de comportamentos físicos e textuais através de cinco opções de execução possíveis para cada um deles. Contudo, como a entrada das informações fica a cargo do escritor, neste momento, ele é quem irá determinar o grau de improvisação que o agente irá possuir.

Podemos pensar nos componentes do planejamento tradicional como algo que os agentes podem usar para guiar seu curso de ação e não como algo que irá ser usado para planejar com antecedência um curso de ação completo. Neste sentido, entendemos que os componentes pré-condição, ação, e pós-condições ou efeitos estão envolvidos na organização de uma apresentação e por este motivo são considerados nas bases de informação como precedência, efeito e características físicas e textos, relacionados à ação. Desta maneira, as bases de informação possuem os formatos descritos nas tabelas 11.1, 11.2 e 11.3, pois elas fornecem o embasamento para a construção da intenção do agente no formato de um *script* abstrato de comportamento, como iremos apresentar na próxima seção.

11.4 Improvisação de Texto: Construção da Intenção como Script Abstrato de Comportamento (SAC)

O módulo da construção da intenção como *script* abstrato de comportamento, como o próprio nome sugere, tem como objetivo criar intenções para guias virtuais envolvidos na apresentação de conteúdos. A construção destas intenções envolve o tipo de improvisação

de texto, relacionada ao curso de ação, e por este motivo, o módulo engloba duas fases: a preparação e a execução da intenção como um SAC.

A fase da preparação da intenção está relacionada ao agente no seu papel de diretor, e a fase de execução está relacionada ao agente no seu papel de ator. Desta maneira, o módulo possui uma arquitetura genérica que se aplica aos diferentes papéis através de algoritmos particulares para cada um dos papéis executados pelos agentes. A arquitetura do módulo é composta por dois submódulos relacionados a construção de uma intenção como um *script* abstrato de comportamento: o escalonador de atividades e o escalonador de comportamento (figura 11.8).

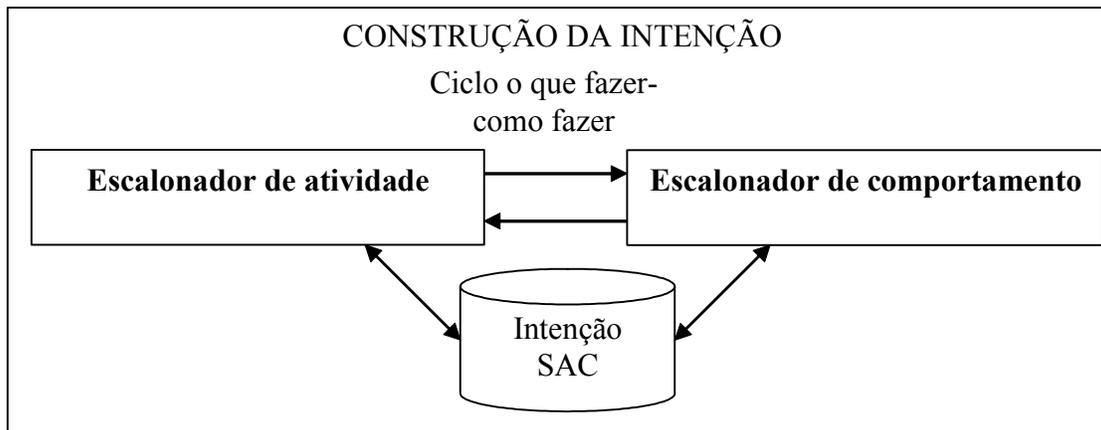


Figura 11.8: Arquitetura genérica do módulo de construção de intenções

O escalonador de atividade constrói a parte da intenção relacionada ao o que o agente deve fazer, e o escalonador de comportamento constrói a parte relacionada a como fazer. O módulo realiza um ciclo de trocas de mensagens entre os dois submódulos para construção da intenção ao nível de preparação ou execução dependendo do papel executado pelo agente. A intenção construída não é um plano diretamente executável, mas sim um plano abstrato contendo indicações de comportamento que devem ser selecionados em tempo de execução. Por este motivo a intenção é chamada de um *script* abstrato de comportamento.

Ao longo desta seção iremos apresentar o nosso entendimento sobre intenções, discutindo a questão de planos como programas versus planos como intenções. Mostraremos uma abordagem para improvisação de texto como satisfação de restrições, os papéis dos agentes e sua relação com os tipos de improvisação de texto, as classes de restrições relacionadas à improvisação de texto e o funcionamento do módulo para os diferentes papéis que o agente pode executar (diretor, ator e diretor-ator) apresentando as arquiteturas e algoritmos específicos.

11.4.1 Visão de Planos como Programas versus Planos como Intenções

Para discutir sobre estas duas visões, primeiramente temos que considerar que o mundo real não é estático e previamente conhecido. Pelo contrário, os ambientes reais são dinâmicos e podem mudar enquanto um agente está raciocinando sobre como alcançar um objetivo e estas mudanças podem invalidar as suposições que um agente se baseia para executar raciocínio. Agentes que habitam ambientes reais e dinâmicos precisam ser receptivos a potenciais objetivos, para isto eles devem raciocinar sobre suas ações e

conhecer quando novos fatos e oportunidades acontecem e se adaptar às situações correntes [POL 92].

Muitos autores [AGR 87] [AGR 88a] [AGR 88b] [AGR 97] [SUC 87] [AND 95] têm considerado como um agente pode usar a situação corrente a fim de agir. Suas abordagens são diferentes do planejamento tradicional e estão relacionadas à improvisação. Todos os autores concordam que a abordagem do planejamento clássico não pode ser aplicada a ambientes dinâmicos e complexos.

Do ponto de vista do planejamento existem duas maneiras de ver um plano. No planejamento clássico um plano é visto como um programa e numa abordagem alternativa um plano é visto como uma intenção.

De acordo com Pflieger e Hayes-Roth [PFL 98] na visão de planos como programas, um plano é um programa executável consistindo de ações primitivas que um agente executa a fim de agir. Logo, o planejamento é um tipo de programação automática e a execução de um plano consiste de uma execução direta. Na outra visão, um plano é um comprometimento que orienta, mas não determina unicamente as ações específicas que um agente executa. Desta maneira, o agente não executa diretamente o seu plano, ele somente executa comportamentos que podem estar mais ou menos consistentes com seu plano.

Os planos como programas determinam diretamente as ações do agentes. Esta visão tem várias limitações como problemas computacionalmente intratáveis, inadequação para um mundo caracterizado por eventos imprevisíveis e necessidade de um plano muito detalhado [AGR 88a]. Na visão alternativa, um plano não determina diretamente as ações do agente. Pelo contrário, um plano é um recurso que um agente pode usar quando irá decidir o que fazer.

Iremos utilizar a visão alternativa de planos como intenções. Entendemos que a intenção de um agente pode ser vista como um tipo plano, capaz de improvisação, voltado para a satisfação de restrições. Na próxima seção iremos abordar as questões de intenção, improvisação e satisfação de restrições.

11.4.2 Intenção, Improvisação e Satisfação de Restrições

Como mencionado anteriormente, alguns autores vêem planos como programas e outros vêem planos como intenções. Nosso trabalho se enquadra na segunda visão. Entendemos que as intenções estão fortemente relacionadas a algum grau de improvisação. E este grau é determinado por restrições que o agente encontra quando está tentando alcançar suas intenções. Por exemplo, um agente (humano neste caso) pode ter a intenção de chegar ao trabalho as 8:30h da manhã. A satisfação desta intenção irá depender de muitos fatores. O agente tem que realizar uma série de atividades antes de chegar ao trabalho, tais como: acordar, tomar café e pegar algum transporte para ir até o trabalho. Algumas destas atividades têm uma certa ordem, mas outras não. Normalmente, as pessoas acordam antes de tomar café. As atividades são realizadas de acordo com as motivações e limitações do agente. Da mesma maneira, cada uma destas atividades pode ter várias possibilidades de execução, por exemplo, um dia o agente pode ir ao trabalho de ônibus ou de carro, mas estas possibilidades também estão restritas a algumas condições. Se o agente está atrasado é melhor ele ir de carro, se o carro não tem gasolina é melhor ir de ônibus, e assim por diante. No decorrer do curso de ação algumas oportunidades podem acontecer e o agente tem que considerá-las.

O objetivo é mostrar que para alcançar suas intenções as pessoas podem ter uma idéia do que fazer e isto está relacionado a vários tipos de limitações e oportunidades que irão ser chamadas de intenções. O agente tem liberdade de improvisar suas ações considerando as restrições presentes no momento. Esta visão está relacionada a abordagem alternativa do planejamento, de ver planos como intenções que indicam não somente os objetivos dos agentes mas algum conjunto de comportamentos possíveis para se alcançar estes objetivos [PFL 98] [PFL 96] [AGR 88a].

De maneira resumida, o agente tem alguma intenção e esta intenção poderia ser descrita como um *script* abstrato de comportamento que indica algumas linhas do que fazer e como fazer algo para alcançar a sua intenção. Estas linhas são abstratas, descrevendo procedimentos gerais do que fazer, elas não descrevem um plano detalhado do que fazer. O agente somente pode escolher o que fazer e como fazer quando ele está em alguma situação. Uma intenção poderia ser entendida como um objetivo que é representado através de um *script* de alto nível que é instanciado com ações concretas de acordo com a situação corrente. O *script* de alto nível e as ações concretas são representadas como improvisações que são realizadas através da satisfação de restrições.

Como mencionado por Chacra [CHA 83] no capítulo 4, a improvisação é algo informal, espontâneo, inspirado na situação atual, enquanto o teatro tradicional é o oposto, ou seja, algo preparado, elaborado e organizado. Embora a improvisação e o teatro tradicional sejam diferentes, Chacra [CHA 83] salienta que os dois são pólos opostos de uma mesma matéria. A diferença entre estes dois pólos é determinada por graus que tornam a apresentação teatral mais ou menos formalizada ou improvisada.

Estes graus podem ser determinados através de diferentes tipos de improvisação como os especificados no capítulo 6, a saber: improvisação de texto, improvisação de personalidade, improvisação do público e improvisação de solução de problemas. O módulo de construção da intenção ou *script* abstrato de comportamento está relacionado ao desenvolvimento da improvisação que ocorre durante um curso de ação, ou seja, a improvisação de texto. Entendemos que este tipo de improvisação irá ser realizado como um problema de satisfação de restrições das limitações e oportunidades dos agentes.

Embora não esteja sendo tratada a questão da personalidade do agente neste momento, cabe salientar que o módulo de construção de intenções poderia incorporar facilmente a improvisação de personalidade. Assim como a improvisação de texto, a improvisação de personalidade poderia ser realizada usando a satisfação de restrições.

Vários problemas da Inteligência Artificial e de outras áreas da Ciência da Computação podem ser vistos como casos especiais do problema de satisfação de restrições (CSP). Alguns problemas são: visão de máquina, manutenção de crenças, escalonamento, raciocínio temporal, planejamento com experimentos genéticos, problemas de satisfação e raciocínio de diagnóstico [KUM 92].

De acordo com Kumar [KUM 92] problemas de satisfação de restrições podem ser descritos com tendo um conjunto de variáveis, um domínio discreto e finito para cada variável e um conjunto de restrições. Cada restrição é definida sobre algum subconjunto do conjunto original de variáveis e restringem a combinação de valores possíveis que um subconjunto de variáveis pode ter.

Dechter e Rossi [DEC 2000] vêem o mesmo problema como uma rede de restrições que consiste de um conjunto de variáveis finitas, cada um associado com um domínio e um conjunto de restrições. Uma solução para este problema é uma atribuição de valores para cada variável, considerando seu próprio domínio, de modo que todas as restrições sejam satisfeitas. Problemas típicos de satisfação de restrição são determinar se uma solução existe, encontrar uma ou mais soluções e encontrar a solução ótima considerando uma função de custo.

Restrição é uma formalização matemática de relacionamentos que ocorrem entre objetos. Por exemplo, perto de ou longe de são restrições que acontecem entre objetos no mundo real [MAR 98].

Restrições podem estar presentes em todos os tipos de atividades diárias. Por exemplo, considere a restrição $T = P - P/10$. Esta restrição poderia ser vista como um total T a ser pago por um produto P com 10% de desconto. De acordo com Marriott e Stuckey [MAR 98] a forma e significado de uma restrição é especificada por um domínio de restrição. Desta maneira, as restrições são escritas em uma linguagem com constantes, funções e relações de restrição. No exemplo acima, a constante é 10, as funções são subtração e divisão e a relação de restrição é a igualdade.

O domínio de restrição especifica a sintaxe de uma restrição, ou seja, especifica as regras para criar restrições em um determinado domínio. O domínio detalha as constantes, as funções e as relações de restrições permitidas.

Como salientado por Kumer [KUM 92] existem vários domínios onde a satisfação de restrições pode ser aplicada. Concordamos com Kumer e iremos explorar aqui um domínio ainda não estudado em satisfação de restrição, que é a improvisação. Entendemos que a improvisação de texto é um tipo de problema de satisfação de restrições e conseqüentemente as intenções dos agentes envolvam improvisação de texto usando satisfação de restrições.

Ao longo destas próximas seções iremos mostrar como os agentes utilizam improvisação de texto para satisfação de restrições. Apresentaremos nossas classes e tipos de restrição, os domínios de restrição e como eles são aplicados aos agentes.

11.4.3 Papéis dos Agentes e Tipos de Improvisação do Texto

O módulo de construção da intenção ou *script* abstrato de comportamento é executado pelo agente quando o mesmo se encontra em qualquer um dos três papéis possíveis: diretor, diretor-ator e ator. Contudo, cada papel implica em um funcionamento diferente do módulo, visto que o diretor é responsável por gerenciar os atores de maneira improvisada, o ator é responsável por improvisar sua performance de acordo com as direções recebidas e o diretor-ator é responsável por realizar as duas coisas ao mesmo tempo. Isto significa que quando o agente está no seu papel de diretor ele é responsável pela fase de preparação da improvisação de texto, quando o agente está no seu papel de ator ele é responsável pela fase de execução da improvisação do texto e por último quando o agente exerce os dois papéis simultaneamente ele é responsável em um momento pela preparação da improvisação do texto e em outro momento pela sua execução.

Dependendo do papel que o agente executa em um determinado momento, ele irá possuir uma intenção relacionada a uma das fases da improvisação de texto. Se o agente é um diretor, ele irá possuir uma intenção relacionada à fase de preparação da improvisação

de texto. Se o agente é ator, ele irá possuir uma intenção relacionada à fase de execução da improvisação de texto e se o agente é diretor-ator, ele irá possuir uma intenção relacionada à fase de preparação e outra intenção relacionada à fase de execução da improvisação do texto. Entretanto, independente do papel que o agente está executando, a improvisação de texto será trabalhada como um processo de satisfação de restrições, sendo que estas restrições variam dependendo do papel que o agente atua em um momento.

11.4.4 Classes e Tipos de Restrições

Na arquitetura existem duas classes de restrições: restrições de ordem de conteúdo e restrições de comportamento. Na classe de restrições de ordem de conteúdo estão todos os tipos de restrições relacionadas à ordem na qual algum conteúdo deveria ser organizado ou apresentado. Na classe de restrições de comportamento estão todos os tipos de restrições relacionados ao processo de selecionar um comportamento apropriado para executar. Durante as próximas seções apresentaremos os tipos de restrições utilizadas e em que classes elas estão presentes.

11.4.5 Funcionamento do Módulo para o Papel de Diretor

Nesta seção iremos descrever a intenção do agente quando ele está executando seu papel de diretor, a arquitetura do módulo e o seu funcionamento para este papel. Para tanto, será mostrado como é abordada a questão da satisfação de restrições e serão apresentados os algoritmos de alto nível relacionados ao módulo.

11.4.5.1 Intenção do Agente no Papel de Diretor

A fim de coordenar os componentes da arquitetura, o diretor tem a sua própria intenção descrita como um *script* abstrato de comportamentos, como mostra a figura 11.9. A primeira tarefa que o diretor realiza, é ativar o seu módulo de aquisição de conhecimento para obter informação sobre o que deve ser apresentado. Após ter obtido informação sobre o que apresentar, o diretor executa o seu módulo de construção da intenção ou SAC, para construir intenções como *scripts* abstratos de comportamento para seus atores. Após construir os *scripts*, o diretor ativa o módulo de ação para enviar os *scripts* aos atores. Neste momento, o diretor passa a aguardar por alguma percepção. O diretor pode receber dois tipos de percepção, uma indicando o fim de uma apresentação e outra indicando a falha em uma apresentação. Se a percepção for de fim, o diretor termina a sua intenção. Se a percepção for de falha, o diretor chama o módulo de improvisação de solução de problemas. Após executar, o módulo de improvisação de solução de problemas envia o *script* gerado para o módulo de ação. Este processo se repete até a percepção indique a finalização da apresentação.

1. Ativa módulo de aquisição de conhecimento
2. Ativa módulo de construção da intenção/SAC
3. Ativa módulo de ação para enviar *scripts* gerados aos atores
4. Aguarda percepção
5. Enquanto percepção é diferente de fim
 - Início
 - 5.1 Ativa módulo de improvisação de solução de problemas
 - 5.2 Módulo de improvisação de solução de problemas envia *script* para o módulo de ação
 - 5.3 Aguarda nova percepção
 - Fim do enquanto
6. Fim da intenção

Figura 11.9: Intenção do agente no seu papel de diretor

De acordo com Wooldridge [WOO 2000] quando o agente adota uma intenção ele se compromete a ela e utiliza um mecanismo chamado de estratégia de comprometimento para determinar quando e como desistir de uma intenção. Wooldridge [WOO 2000] apresenta três estratégias de comprometimento que são normalmente discutidas na literatura de agentes racionais:

- comprometimento cego: o agente continua a manter uma intenção até que ele acredite que a intenção tenha sido alcançada;
- comprometimento convergente de direcionamento único: agente continua a manter uma intenção até que ele acredite que a intenção tenha sido alcançada ou que não é mais possível atingir a intenção;
- comprometimento mente-aberta: agente mantém a intenção enquanto ele acredita que ela ainda é possível.

Em relação a sua própria intenção, descrita na figura 11.9, o diretor possui um comprometimento cego. Isto significa que ele não irá reconsiderar sua intenção, ele a mantém até o momento que acredita que ela foi alcançada, ou seja, chegou ao fim da intenção, no caso da figura 11.9.

Contudo, em relação à intenção que o diretor gera para seus atores, ele possui um comprometimento mente-aberta, ou seja, se ele acredita que não é mais possível executar a intenção, ele irá reconsiderar esta intenção. Esta intenção será reconsiderada através do módulo de improvisação para solução de problemas. A reconsideração da intenção está descrita no passo 5.1 e 5.2 da intenção descrita na figura 11.9.

A reconsideração da intenção realizada pela improvisação para solução de problemas implementa uma das características que a função filtro de uma arquitetura BDI deve possuir: adotar novas intenções, ou para atingir as intenções existentes ou para explorar novas oportunidades. Como veremos mais adiante, o módulo para improvisação para solução de problemas incorpora um tipo de função filtro.

11.4.5.2 Restrições do Diretor para Construção do Script Abstrato de Comportamento

De acordo com Spolin [SPO 63] e Brockett [BRO 76], o diretor tem que fornecer direções de como alguma coisa pode ser realizada. Ele tem que dar um problema para os atores. Este problema é descrito abstratamente e a maneira como os atores irão solucionar este problema é individual. Cada ator tem a sua própria maneira de resolver um problema. Não existe uma maneira previamente conhecida e considerada como correta.

No papel de diretor, o agente usa conhecimento sobre a apresentação para construir um problema para cada ator, chamado aqui de intenção do ator. Assim como no teatro, esta intenção irá ser informada na forma de um *script*. Este *script* é dinâmico, pois não dita o comportamento do ator e o ator irá escolher sua performance de acordo com o seu ambiente em um certo momento durante a execução. Os atores irão instanciar os *scripts*. Desta maneira, os *scripts* fornecem classes de ação relacionadas a alguma atividade e os atores têm que escolher que ações executar a cada momento, dependendo das restrições relacionadas tanto à ação quanto ao ator.

Para construir intenções na forma de *scripts* para seus atores, o diretor utiliza improvisação de texto como um processo de satisfação de restrições. As restrições que o diretor manipula estão presentes na classe de restrições de ordem de conteúdo. Esta classe está relacionada a ordenação do conteúdo de apresentação. Os tipos de restrições nesta classe são:

- precedência: pré-condições relacionada à ordem de alguma execução;
- efeitos: quais efeitos a execução de alguma atividade irá trazer. A ativação de um efeito traz a necessidade de satisfação de uma nova precedência;
- estado do *script*: indica se o *script* está vazio ou não.

O domínio de restrição neste caso é composto por constantes que indicam, vazio, não vazio, nenhum e fim e as relações de restrições são igualdade (=), diferença (!) e existência (\exists). Figura 11.10 mostra alguns exemplos de ações na construção do *script* e suas restrições relacionadas.

Ação	Exemplo de Restrição
Procure por uma atividade cuja precedência é nenhuma	estado do <i>script</i> é igual a vazio
Armazene efeito intermediário Armazene precedência no <i>script</i> Chame o escalonador de comportamento Armazene o efeito no <i>script</i>	estado do <i>script</i> é diferente vazio e existe atividade cuja precedência é igual ao efeito atual
Atribua fim à precedência	efeito é igual a fim

Figura 11.10: Exemplos de ações e restrições relacionadas no módulo de construção de *scripts* para o papel de diretor

11.4.5.3 Diretor como Autor

O diretor também é um autor, pois o escritor humano informa as atividades com as suas precedências e conteúdos para o diretor, e o diretor tem que organizar estas atividades em ordem para informar a um ator específico o que deve ser feito. A ordem da apresentação irá depender da ordem que o escritor informou ao diretor. Algumas vezes o escritor pode deixar alguma ordem em aberto, por exemplo, existem três lugares diferentes para se apresentar, mas o escritor não especifica uma ordem entre eles. Desta maneira, o diretor pode deixar esta ordem em aberto para o ator decidir sua execução.

Além de guiar o que o ator tem que fazer, o diretor pode guiar como uma atividade irá ser executada. O diretor informa ao ator quais classes de ações podem ser relacionadas a algum conteúdo, e os atores irão escolher de acordo com suas restrições, qual a melhor ação para ser executada em um determinado momento.

11.4.5.4 Arquitetura do Módulo para o Papel de Diretor

Basicamente a construção do *script* é dividida em dois módulos. O primeiro está relacionado ao que o ator deve fazer e é chamado de escalonador de atividades. O segundo está relacionado a como os atores executam suas atividades e é chamado de escalonador de comportamento. Quando o agente está no papel de diretor, o módulo de construção da intenção está executando a fase de preparação da improvisação do texto. A figura 11.11 mostra estes dois módulos.



Figura 11.11: Arquitetura de módulo da construção do script para o papel de diretor

11.4.5.4.1 Representação das Intenções como Scripts Abstratos de Comportamento para Agente Executando Papel de Ator

A representação das intenções como *scripts* abstratos de comportamento é baseado no modelo de regras de produção. De acordo com Stefik [STE 95], cada regra de produção tem duas partes, chamadas de parte se e parte então. A parte se de uma regra consiste de condições a serem testadas. Se todas as condições da parte se de uma regra são verdadeiras, as ações da parte então da regra são executadas. A parte então destas regras consiste de ações a serem executadas. A diferença fundamental entre a nossa representação e as regras de produção tradicionais é que as ações na parte então são comportamentos abstratos, ou ações de alto nível, que são transformadas em ações primitivas durante a improvisação de texto que ocorre na fase de execução.

O mapeamento entre regras se..então tradicionais para estrutura da intenção como *script* abstrato de comportamento é mostrado na figura 11.12.

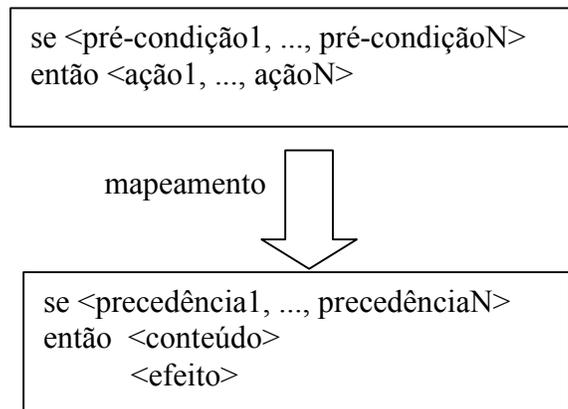


Figura 11.12: Transformação de uma regra se..então em uma estrutura de intenção como *script* abstrato de comportamento

Precedência é a precedência de alguma atividade. Conteúdo é uma indicação de um processo que deve ser chamado no ator para escolher que conteúdo e ação específica deve ser executado. O ator tem que usar suas restrições para escolher quais conteúdos e ações executar, porque o diretor somente informa alguma ordem e classes de ações para o ator. Efeito é o efeito ou efeitos relacionados à execução de uma atividade. A ativação de um efeito influencia na satisfação de uma ou mais precedências.

Em alguns casos, pode existir mais de uma precedência que seja satisfeita em um determinado momento. Quando um ator está executando seu *script* e isto acontece, ele deve escolher qual é a melhor opção de acordo com as suas restrições (limitações e oportunidades). O ator tem que improvisar a execução do seu *script* considerando as restrições de uma determinada situação.

Como se pode observar, a intenção representada como um *script* abstrato de comportamento é uma descrição abstrata do que e de como o ator irá executar uma determinada atividade. O diretor não dita o comportamento do ator. Ele fornece informações gerais e deixa o ator livre para agir de acordo com as suas restrições. O diretor e o ator estão trabalhando em conjunto para apresentar algum conteúdo ao usuário.

11.4.5.4.2 Algoritmos do Escalonador de Atividades e do Escalonador de Comportamentos para a Fase de Preparação da Improvisação de Texto

O diretor inicia o processo de construção da intenção como um *script* estruturado em regras de decisão abstratas através do escalonador de atividades. O escalonador de atividades funciona da seguinte maneira: procura por curso de ação igual ao objetivo informado; enquanto o efeito de uma atividade do curso de ação é diferente de fim, o diretor procura por atividades, que ainda não estejam na intenção, através de suas precedências. Ao encontrar, o diretor armazena a precedência na intenção, chama o escalonador de comportamento para armazenar a parte relativa à ação e armazena o efeito que será gerado após a execução da ação. Ao finalizar a construção da intenção como *script*, o diretor relaciona um ator ao *script* e grava a intenção na biblioteca de *scripts*. O algoritmo de alto nível do escalonador de atividades pode ser visualizado na figura 11.13.

Para cada atividade que o diretor encontra uma precedência igual ao efeito atual, ele grava a atividade na lista de atividades da intenção. Antes de inserir uma nova atividade na intenção, o diretor verifica se ela já não existe na intenção. Isto deve ser feito devido à possibilidade de existir precedências compostas, ou seja, que possuem mais de um efeito associado. Quando o diretor encontra uma precedência composta, o primeiro efeito igual a uma das precedências ocasiona o armazenamento da atividade na intenção. Desta maneira, os demais efeitos que ativariam esta atividade não podem ocasionar uma nova gravação da mesma atividade. Por este motivo, o diretor antes de gravar uma atividade na intenção verifica se ela já foi previamente armazenada.

O passos 3 e 4 do algoritmo, apresentado na figura 11.13, têm como objetivo gravar na intenção que será gerada para o ator uma regra para identificar que ocorreu uma falha durante a execução da intenção. Esta regra diz que se nenhuma precedência é ativada então deve-se informar ao diretor, através de uma mensagem, que ocorreu uma falha e se solicitar uma solução para a mesma passando para o diretor a requisição do usuário que ocasionou a falha. O efeito desta regra é deixar o ator em um estado ocioso até que ele receba novas instruções do diretor.

1. procura por curso de ação igual ao objetivo informado
2. enquanto efeito for diferente de fim para atividades do curso de ação
 - início
 - 2.1 se intenção está vazia
 - início
 - 2.1.1 insere atividade inicial na lista de atividades da intenção
 - 2.1.2 procura atividade inicial cuja precedência seja nenhuma
 - 2.1.3 anexa atividade na intenção/SAC – parte relativa a(s) precedência(s)
 - 2.1.4 chama escalonador de comportamento
 - 2.1.5 armazena efeito na intenção/SAC
 - 2.1.6 atualiza variável de efeito com efeito da atividade
 - fim
 - 2.2 senão
 - início
 - 2.2.1 enquanto existir atividade cuja precedência seja igual ao efeito atual
 - início
 - 2.2.1.1 se atividade não existe na lista de atividades então
 - início
 - 2.2.1.1.1 insere atividade na lista de atividades da intenção
 - 2.2.1.1.2 armazena efeito da atividade na lista de efeitos intermediários
 - 2.2.1.1.3 anexa atividade na intenção/SAC – parte relativa a(s) precedência(s)
 - 2.2.1.1.4 chama escalonador de comportamento
 - 2.2.1.1.5 armazena efeito na intenção/SAC
 - fim
 - 2.2.1.2 procura próxima atividade
 - fim enquanto
 - 2.2.2 atualiza efeito atual com primeiro efeito intermediário da lista de atividades
 - fim
 - fim enquanto
3. Anexa atividade relativa a identificação de falha na intenção/SAC – parte relativa a(s) precedência(s)
4. Armazena efeito na intenção SAC
5. Relaciona ator para intenção/SAC
6. Grava intenção/SAC na biblioteca de *scripts*

Figura 11.13: Algoritmo de alto nível do escalonador de atividades para o papel do diretor

O algoritmo de alto nível do escalonador de comportamento pode ser visualizado na figura 11.14.

- | |
|---|
| <ol style="list-style-type: none"> 1. anexa na intenção/SAC procura_conteúdo(atividade) 2. anexa na intenção/SAC procura_características_fisicas(atividade) |
|---|

Figura 11.14: Algoritmo de alto nível do escalonador de comportamento para o papel de diretor

O escalonador de comportamento tem como objetivo gravar na intenção comportamentos abstratos relativos às possíveis falas e gestos do agente ator. Ao gravar *procura_conteúdo(atividade)* na intenção, o diretor está deixando a cargo do ator a escolha do conteúdo que será apresentado ao usuário. Ao gravar *procura_características_fisicas(atividade)* na intenção o diretor também deixa a cargo do ator a escolha, no momento da execução, do comportamento físico que irá executar. A variabilidade de comportamento está relacionada às opções disponíveis nas bases de informações de atividades e conhecimento. Desta maneira, o ator recebe do diretor uma intenção como um script abstrato de comportamentos a serem executados, e fica a cargo do ator, com base nas suas restrições, escolher as melhores opções de execução.

11.4.6 Funcionamento do Módulo para o Papel de Ator

Quando o agente está executando o papel de ator, o módulo da construção da intenção como SAC está relacionado à fase da execução da improvisação de texto. A arquitetura permanece a mesma do módulo para o papel do diretor, contudo o escalonador de atividade e o escalonador de comportamento estão relacionados à definição de uma forma de execução para a intenção fornecida pelo diretor. Desta maneira, os algoritmos do escalonador de atividade e do escalonador de comportamento devem ser diferentes, pois eles têm como objetivo instanciar os comportamentos contidos no *script* abstrato de comportamento na forma de um comportamento concreto executável pelo ator.

11.4.6.1 Intenção do Agente no Papel de Ator

A intenção do agente no papel de ator é a intenção gerada como *script* abstrato de comportamento pelo diretor, ou seja, é um conjunto de regras de alto nível. O ator deve seguir as regras construídas pelo diretor improvisando sua execução de duas maneiras: através da seqüência de atividades a ser executada, quando houver a possibilidade de executar mais de uma atividade em um determinado momento, e através da escolha dos comportamentos a serem executados. Inicialmente, o agente ator pode improvisar dois tipos de comportamentos, verbais e físicos. Os comportamentos verbais são improvisados através das diferentes opções de texto do conteúdo, e os comportamentos físicos são improvisados através das diferentes características físicas relacionadas à atividade. Um exemplo genérico da intenção de um ator pode ser visualizado na figura 11.15.

Na intenção descrita na figura 11.15, a atividade1 é a atividade inicial, e a atividade4 é a atividade final. O ator pode escolher a ordem de execução entre a atividade2 e a atividade3, pois ambas têm a mesma precedência. Caso a precedência atual seja

diferente de todas as precedências anteriores, é enviada uma mensagem para o diretor informando que ocorreu uma falha e passando a precedência que ocasionou a falha. A precedência neste caso indica um conteúdo que foi solicitado para apresentação e que não estava contemplado na intenção atual do agente.

Como o objetivo do agente ator é satisfazer seu usuário apresentando conteúdo por ele desejado, ao ocorrer uma falha na intenção, o ator solicita ao diretor uma reconsideração da intenção. Isto ocorre porque o agente ator possui um comprometimento do tipo mente aberta em relação a sua intenção. Este tipo de comprometimento foi adotado no ator para que ele pudesse estar receptivo aos novos objetivos e oportunidades requeridas pelo seu usuário.

```

1. Se precedência1 então
  início
  1.1 procura_conteúdo(atividade1)
  1.2 procura_características_físicas(atividade1)
  1.3 efeito1
  fim
2. se precedência2 == efeito 1 então
  início
  2.1 procura_conteúdo(atividade2)
  2.2 procura_características_físicas(atividade2)
  2.3 efeito2
  fim
3. se precedência3 == efeito 1 então
  início
  3.1 procura_conteúdo(atividade3)
  3.2 procura_características_físicas(atividade3)
  3.3 efeito3
  fim
4. se precedência4 == efeito2 e efeito3 então
  início
  4.1 procura_conteúdo_(atividade4)
  4.2 procura_características_físicas(atividade4)
  4.3 efeito4
  fim
5. se precedência diferente das anteriores então
  5.1 envia_mensagem_falha_diretor(precedência)

```

Figura 11.15: Exemplo genérico da intenção de um ator

11.4.6.2 Restrições do Ator para Construção da Execução do Script Abstrato de Comportamento

As restrições que o agente ator opera estão relacionadas à escolha da ordem de execução das atividades e a escolha do conteúdo da atividade e da característica física da atividade que o agente deve executar. As restrições relacionadas à ordem de execução estão presentes na classe de restrições de ordem de execução e possuem os mesmos tipos de

restrição apresentadas para o papel do diretor. O domínio de restrição é estendido para incorporar as constantes operando e falha para indicar um estado do *script*. A figura 11.16 apresenta alguns exemplos de ações e suas restrições relacionadas para o papel de diretor.

Ação	Exemplo de Restrição
Seleciona atividade cuja precedência é satisfeita	estado do script é igual a operando
Ativa módulo de improvisação de solução de problemas	estado do script é igual a falha

Figura 11.16: Exemplos de ações e restrições de ordem relacionadas no módulo de construção de *scripts* para o papel de ator

No momento de escolher o conteúdo e a característica física de uma atividade, o agente ator irá improvisar dois tipos de comportamento, verbais e físicos. Desta maneira, as restrições sobre as quais os atores operam estão relacionadas à classe de restrições de comportamento. Estas restrições estão relacionadas à função do escalonador de comportamento que escolhe uma instância de comportamento verbal e físico para ser executado. O tipo de restrição nesta classe é o estado da lista de comportamentos, que indica se a lista de comportamentos está vazia ou não. O domínio de restrição, neste caso, é composto por constantes que indicam vazio, não vazio, as relações de restrições são igualdade (=), diferença ou negação (!) e existência (\exists). Figura 11.17 mostra alguns exemplos de ações na seleção do comportamento e suas restrições relacionadas.

Ação	Exemplo de Restrição
Recupera lista de comportamento	estado da lista é igual a vazia e não existente
Recupera lista de comportamento, mas não coloca último comportamento	estado da lista é igual a vazia e existente
Seleciona comportamento da lista	estado da lista é diferente de vazia

Figura 11.17: Exemplos de ações e restrições de comportamento relacionadas no módulo de construção de *scripts* para o papel de ator

Em virtude do agente ator poder ser representado através de um personagem animado, seria interessante que ele possuísse outros fatores que influenciasses na sua improvisação de comportamento, como personalidade, emoção, etc. Porém, o estudo da personalidade e emoções não está no escopo desta tese. O agente poderia possuir traços de personalidade e emoções e estes traços poderiam influenciar na improvisação de

comportamento através da satisfação de suas restrições. Este é um dos trabalhos futuros desta tese.

11.4.6.3 Arquitetura do Módulo para o Papel de Ator

A arquitetura do módulo para o papel de ator pode ser visualizada na figura 11.18. O escalonador de atividades obtém informação da intenção como SAC e seleciona uma ordem de execução das atividades. O escalonador de comportamento recebe uma requisição do escalonador de atividades para selecionar os melhores comportamentos para executar considerando a intenção atual. Após selecionar os comportamentos, o escalonador envia uma mensagem para o escalonador de atividades que ativa o módulo de ação para executar as instâncias de comportamento selecionadas. Ao contrário da construção da intenção para o papel do diretor, o escalonador da atividade e escalonador de comportamento somente acessam a intenção como SAC, eles não gravam informações na mesma.

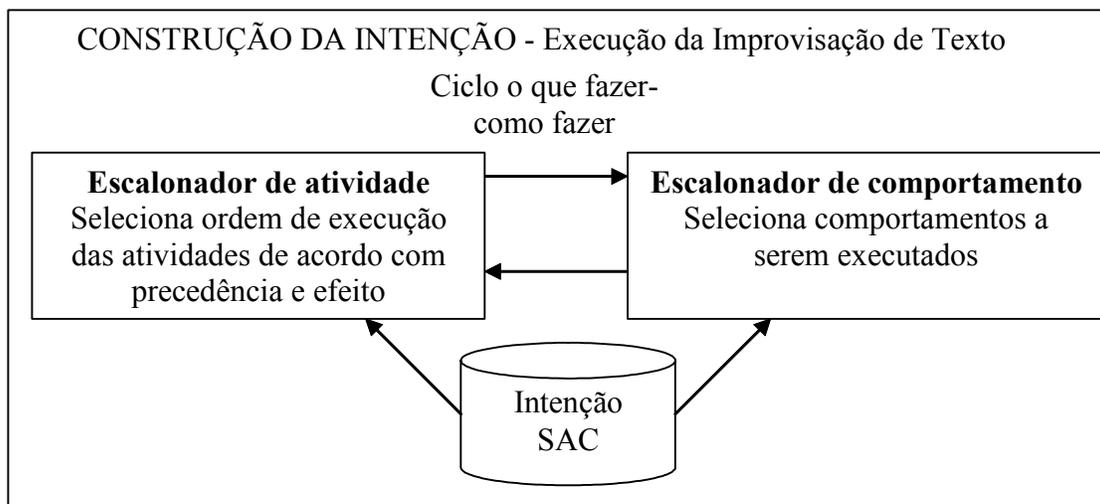


Figura 11.18: Arquitetura do módulo de construção da intenção para o papel de ator

11.4.6.3.1 Algoritmos do Escalonador de Atividades e do Escalonador de Comportamentos para a Fase de Execução da Improvisação de Texto

No caso do ator, o módulo de escalonamento da atividade recebe a intenção como SAC gerada pelo diretor, seleciona a atividade cuja precedência é satisfeita e executa a ação. A ação da intenção requisita a procura dos comportamentos apropriados através da ativação do escalonador de comportamento. O algoritmo pega o retorno do escalonador de comportamento e envia para o módulo da ação para ser executado. Logo após, ativa o efeito relacionado a esta atividade. Caso exista mais de uma atividade relacionada a uma precedência, o escalonador de atividade utiliza uma função para escolha da atividade a ser executada. Se for a primeira vez que esta função é executada, ela escolhe a atividade de maneira randômica. Após escolher uma atividade; guarda as atividades já realizadas e a função na próxima execução considera somente as atividades ainda não apresentadas. A escolha é sempre realizada de maneira randômica, mas o conjunto de atividades vai variando de acordo com a execução das atividades. Caso a atividade escolhida possua precedência composta, ou seja, é necessário mais de um efeito para a sua ativação, o escalonador de atividade ativa a função de solução de precedência composta. Esta função

verifica se as demais componentes da precedência existem na lista de efeitos gerados. Se elas existirem, verifica, para cada uma das componentes se elas possuem outras atividades relacionadas. Caso elas não possuam mais atividades, a função retira as componentes da lista de efeitos gerados; no caso contrário as componentes permanecem na lista de efeitos gerados.

Após ter recebido o conteúdo e a característica particular selecionada pelo escalonador de atividade, o módulo da ação executa a atividade e o agente fica esperando por uma percepção do usuário. O algoritmo de alto nível do escalonador de atividades pode ser visualizado na figura 11.19.

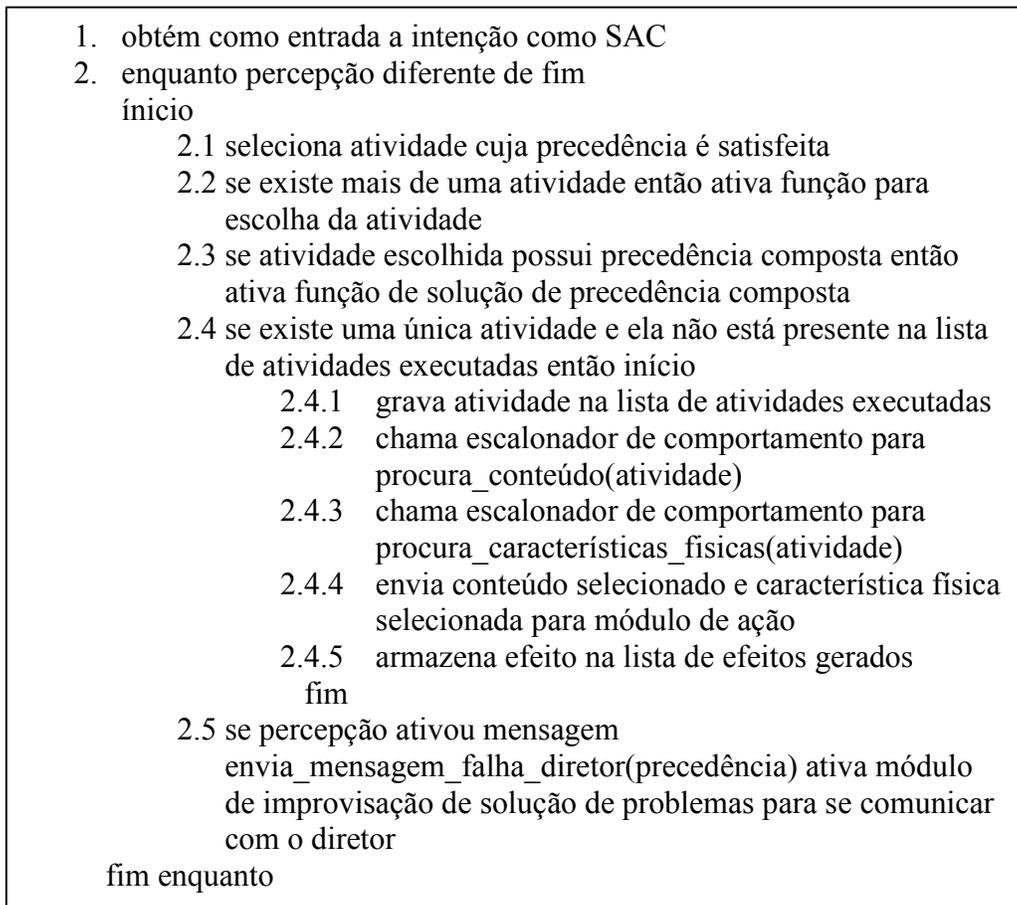


Figura 11.19: Algoritmo de alto nível do escalonador de atividades para o papel de ator

O escalonador de comportamento recebe o conteúdo que tem que procurar e verifica se é conteúdo de texto ou características físicas, a fim de identificar a base de informação que deve ser acessada. Logo após, executa uma função para selecionar o comportamento apropriado. A função para escolha improvisada do comportamento segue o mesmo padrão da função de escolha da atividade. Primeiramente a função seleciona randomicamente entre todos os comportamentos possíveis. O comportamento selecionado é retirado do conjunto e na próxima ativação da função escolhe randomicamente entre os restantes. Quando o conjunto é vazio inicia o processo novamente, mas desconsidera o último comportamento executado. O escalonador de comportamento utiliza uma lista de conteúdos e uma lista de

características físicas para cada atividade, para da próxima vez que a atividade for realizada, escolher um comportamento diferente do que foi executando anteriormente, contribuindo assim para a ilusão de vida do agente. O algoritmo de alto nível do escalonador de comportamento pode ser visualizado na figura 11.20.

Esta versão da função de escolha da atividade e função da escolha de comportamento é uma versão simplificada e pode ser entendida como estando muito próxima do extremo da improvisação mencionado por Chacra [CHA 83], pois utiliza uma função randômica para seleção das atividades e comportamentos. Esta função não é completamente aleatória, pois o conjunto de entrada de seleção é modificado na medida em que a função é executada. Um aperfeiçoamento desta função está relacionado à inclusão dos fatores de personalidade e emoções nos agentes.

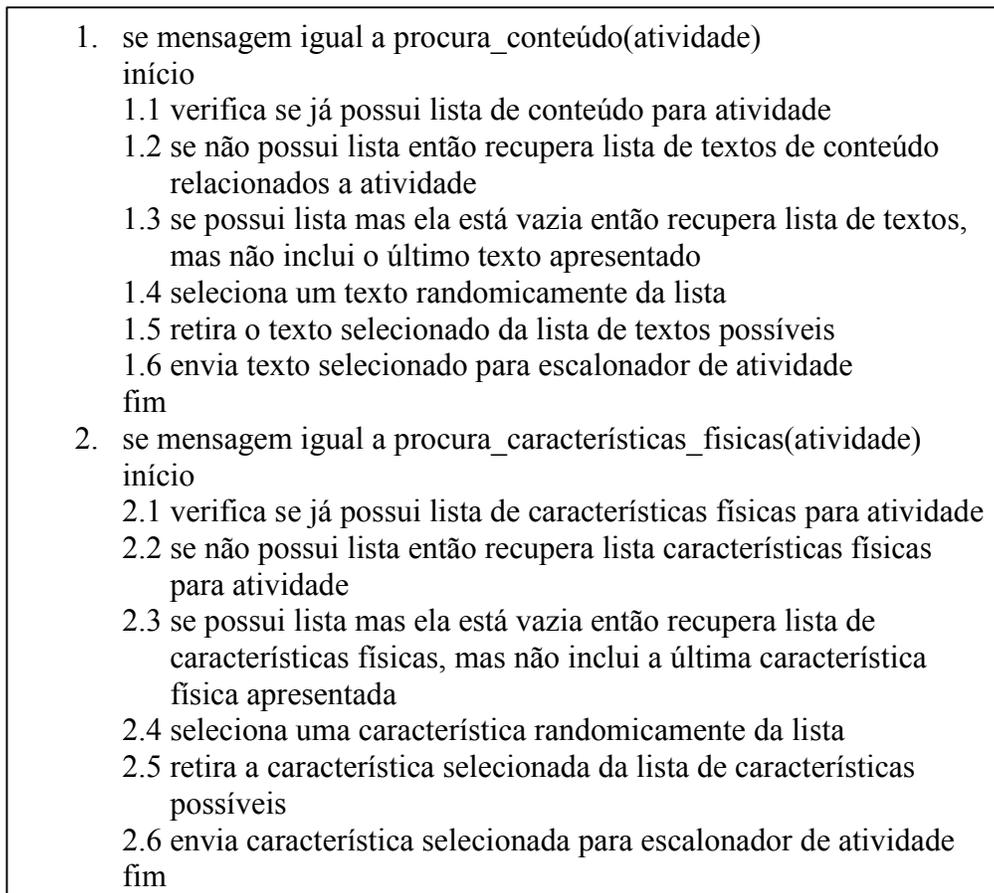


Figura 11.20: Algoritmo de alto nível do escalonador de comportamento para o papel de ator

11.4.7 Funcionamento do Módulo para o Papel de Diretor-Ator

Quando o agente está executando o papel de diretor-ator, ele possui duas intenções, uma intenção ao nível de preparação da intenção como SAC, igual a apresentada na figura 11.4.5.1.1, e uma intenção no nível de execução da intenção como SAC, semelhante a apresentada na figura 11.9. A intenção do diretor-ator segue o mesmo padrão em relação ao comprometimento da intenção como diretor e como ator. Desta maneira, quando está

executando a intenção relacionada ao diretor, o agente possui um comprometimento cego e quando está executando a intenção relacionada ao ator, ele possui um comprometimento mente aberta.

O agente inicia executando o papel de diretor e constrói uma intenção como um SAC. Após gerar o *script* e destinar a ele próprio, o agente interrompe a execução da intenção do diretor e passa a executar a intenção como SAC no papel de ator. Deste modo, o agente diretor-ator irá executar os algoritmos de escalonamento da atividade e escalonamento de comportamento relacionado ao papel que está executando em um determinado momento. Isto significa que quando estiver executando o papel de diretor ele estará manipulando os algoritmos descritos nas figuras 11.13 e 11.14 e quando estiver executando o papel de ator estará manipulando os algoritmos descritos nas figuras 11.19 e 11.20.

Neste momento estamos considerando um caso mais simples de diretor-ator, onde o agente é diretor de si próprio e não de outros agentes. O caso de um diretor-ator ser diretor de outros agentes envolve uma série de controles em relação as requisições que um diretor pode receber de outros atores enquanto está executando seu papel de ator. Uma possibilidade seria fazer com que o agente atendesse as requisições de outros atores quando estivesse no seu estado ocioso de ator ou quando tivesse terminado sua apresentação de conteúdos. Contudo isto parece sobrecarregar o agente e tornar o sistema mais lento, por este motivo, optamos por examinar esta questão em um outro momento, onde poderemos fazer comparações entre os diferentes modos de operação e refletir sobre este assunto mais profundamente.

11.5 Improvisação para Solução de Problemas

O módulo de improvisação para solução de problemas funciona de maneira diferente para cada um dos papéis que o agente pode executar. Quando o agente está executando o papel de diretor, o módulo tem como objetivo improvisar através de analogia por similaridade uma solução para uma falha do agente. Quando o agente está executando o papel de ator, o módulo tem como objetivo requisitar ao agente diretor a solução de um problema identificado. Nas próximas seções será descrito o funcionamento do módulo para cada um dos papéis dos agentes.

11.5.1 Funcionamento do Módulo para o Papel de Diretor

Após o diretor construir uma intenção como um SAC para um ator, ele fica aguardando por uma percepção. Uma percepção é composta por tipo, identificador do agente, e descrição de um objeto, como mostra a tabela 11.4.

Tabela 11.4: Estrutura de uma percepção do agente diretor

Tipo	Identificador do Agente	Descrição do Objeto
------	-------------------------	---------------------

Uma percepção, enviada por um ator, pode ser de dois tipos: fim e falha. Se a percepção for do tipo fim, através do identificador do agente o diretor sabe qual intenção que foi finalizada e o campo descrição do objeto é vazio. Se a percepção for do tipo falha, a estrutura da percepção estará completa, contendo o identificador do agente e a descrição do objeto. Neste caso, o identificador do agente indica qual ator gerou a falha e

conseqüentemente qual intenção deve ser reconsiderada. O campo descrição do objeto irá conter informação sobre os usos possíveis e características particulares do objeto que ocasionou uma falha na intenção do ator.

Ao receber uma percepção, o diretor, através do seu módulo de percepção, verifica se a percepção é diferente de fim. Se a percepção for diferente de fim, o módulo de percepção grava na base de informação de falha o identificador do agente que ocasionou a falha e envia para o módulo de improvisação para solução de problemas a descrição do objeto que ocasionou a falha.

A arquitetura do módulo de improvisação para solução de problemas pode ser visualizada na figura 11.21. A arquitetura da figura 11.21 é um detalhamento da arquitetura descrita na seção 10.3.2.3 e apresentada na figura 9.6, pois contém a especificação das crenças do agente para a arquitetura de guias virtuais. As crenças dos agentes estão contidas nas bases de conhecimento adquiridas no módulo de aquisição de conhecimento. Desta maneira, as crenças dos agentes são objetos descritos nas bases de conhecimento de curso de ação, de atividades e de conteúdos.

Os submódulos identificação do tipo de improvisação, construção da analogia e transformação da analogia em intenção funcionam exatamente como descrito nas seções 10.3.2.3.1, 10.3.2.3.2, 10.3.2.3.3 respectivamente. Como os algoritmos descritos para estes submódulos são genéricos e independentes de arquitetura, nesta seção será realizada uma instanciação destes algoritmos para a arquitetura de guias virtuais.

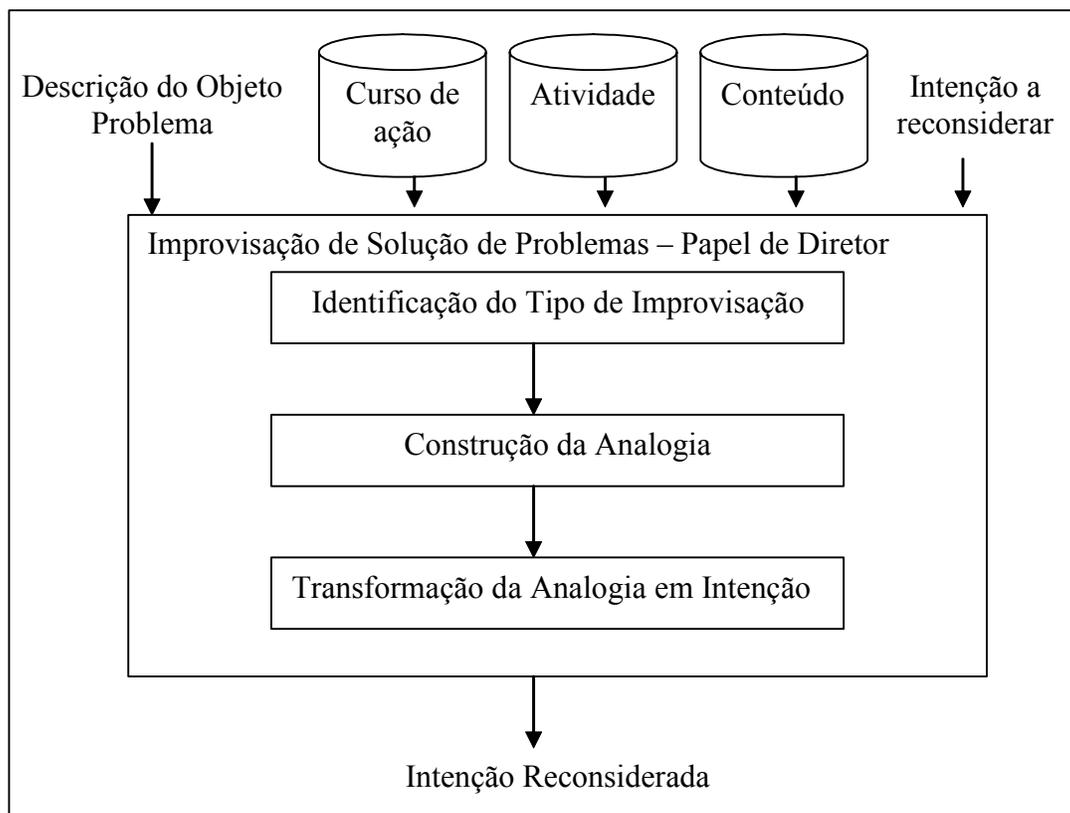


Figura 11.21: Arquitetura do módulo de improvisação para solução de problemas para o papel de diretor

Antes de iniciar a instanciação dos algoritmos é necessário identificar na arquitetura de guias virtuais quais são os objetos meta e objetos meios. Os objetos meta são os objetos contidos na base de curso de ação. Os objetos meio são os objetos contidos nas bases de atividades e de conteúdos.

11.5.1.1 Submódulo de Identificação do Tipo de Improvisação

O algoritmo de alto nível do submódulo de identificação do tipo de improvisação para a arquitetura de guias virtuais é apresentado na figura 11.22.

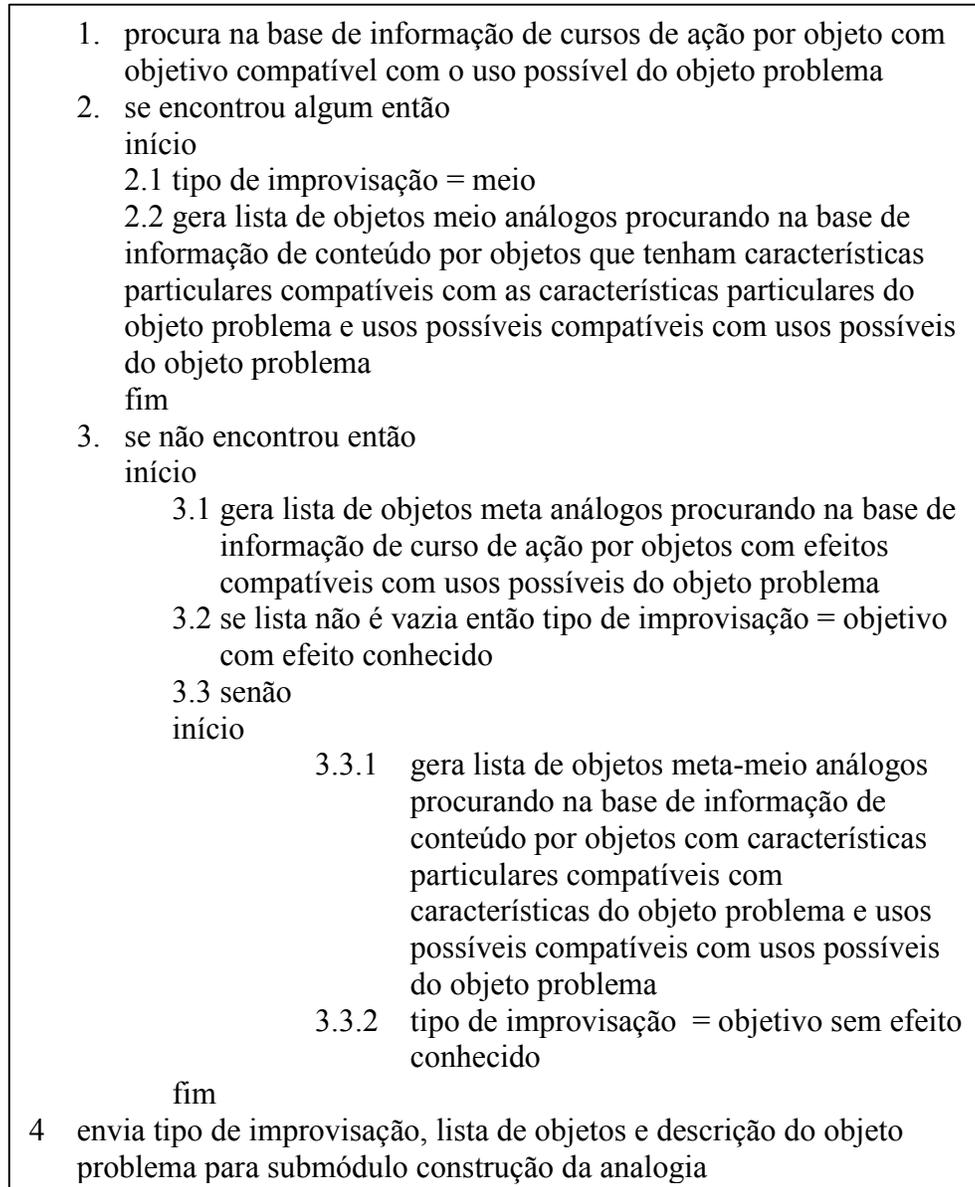


Figura 11.22: Algoritmo de alto nível da identificação do tipo de improvisação para a arquitetura de guias virtuais

11.5.1.2 Submódulo Construção da Analogia

O submódulo construção da analogia recebe como entrada a descrição do objeto problema, a indicação do tipo de improvisação de solução de problemas que deve ser realizada e uma lista das possíveis fontes análogas ao objeto problema. Com base nestas informações, o submódulo através de analogia por similaridade seleciona a fonte análoga mais semelhante ao objeto problema.

Para se calcular o grau de similaridade e diferença, deve-se considerar os atributos que descrevem o alvo, no caso o objeto problema. Os atributos que descrevem o objeto problema são seus usos possíveis e suas características particulares. As características particulares de um objeto problema para a arquitetura de guias virtuais são um subconjunto das características particulares que descrevem os objetos contidos na base de informação de conteúdos. Este subconjunto é composto por tipo, área e subárea. Assim como as características particulares do conteúdo podem ser estendidas dependendo do domínio que se deseja modelar, o conjunto de características particulares do objeto problema também pode ser estendido. O número de características relevantes vai depender do número de características particulares do conjunto. Desta maneira, ao se estender o conjunto de características particulares, o número de características relevantes também pode mudar de acordo com o domínio.

Se o tipo de improvisação é de meio, a similaridade s será calculada considerando os usos possíveis e as características particulares tipo, área e subárea da lista de meios análogos. A diferença d será calculada entre as m características de usos possíveis e particulares do objeto problema e a similaridade s calculada para os objetos fonte análogos da lista de meio análogos. O número de características relevantes para esta analogia será dois. Se o tipo de improvisação é de objetivo sem efeito conhecido a similaridade e diferença será calculada da mesma maneira, mas será aplicada aos objetos fonte análogos da lista de meta-meio objetos.

Se o tipo de improvisação é de objetivo com efeito conhecido, a similaridade s será calculada considerando os efeitos da lista de objetos meta análogos. A diferença d será calculada entre os m usos possíveis do objeto problema e a similaridade s calculada para os objetos fonte análogos da lista de meio análogos. O número de características relevantes para esta analogia será um.

O algoritmo de alto nível do submódulo de construção da analogia para a arquitetura de guias virtuais é apresentado na figura 11.23.

11.5.1.3 Submódulo Transformação da Analogia em Intenção

Este submódulo recebe como entrada o tipo de improvisação e um objeto que é a analogia mais apropriada para a descrição do objeto problema e gera como saída uma descrição de intenção. Se o tipo de improvisação é de meio ou objetivo sem efeito conhecido, o objeto selecionado pela construção da analogia é um objeto da base de informação de conteúdo. No caso contrário, o objeto selecionado é um objeto da base de informação de curso de ação.

Se o tipo de improvisação é de meio ou de objetivo sem efeito conhecido, o submódulo atualiza a intenção atual para que ela contenha uma regra relacionada ao objeto que ocasionou a falha.

Se o tipo de improvisação é de objetivo com efeito conhecido, é necessário gerar uma nova intenção, neste caso o submódulo requisita ao módulo de construção da intenção do diretor a construção de uma intenção. Antes de enviar a requisição para o módulo de construção da intenção, o submódulo recupera da base de informação de falha o identificador do agente que ocasionou a falha e retira o relacionamento do ator para a sua intenção atual, pois a mesma será substituída pela nova intenção. O submódulo passa o identificador do agente e o campo objetivo do objeto selecionado para o módulo de construção da intenção. O módulo de construção da intenção gera a intenção e a envia para o submódulo de transformação da analogia em intenção.

Após ter gerado uma atualização ou uma nova intenção para um ator, o submódulo de transformação da analogia em intenção envia para o módulo de ação a intenção gerada para solucionar a falha, e o módulo de ação envia a intenção para o ator que requisitou a falha. O algoritmo de alto nível do submódulo de transformação da analogia em intenção para a arquitetura de guias virtuais é apresentado na figura 11.24.

1. se tipo improvisação igual a meio então
 - 1.1 enquanto lista de objetos meio análogos é diferente de vazio início
 - 1.1.1 calcula similaridade s considerando características particulares de tipo, área e subárea dos objetos da lista de meio análogos e usos possíveis
 - 1.1.2 calcula diferença d entre m características particulares que descrevem o objeto problema e a similaridade s
 - 1.1.3 $r = 2$
 - 1.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 1.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 fim_enquanto
2. se tipo improvisação igual a objetivo com efeito conhecido então
 - 2.1 enquanto lista de objetos meta análogos é diferente de vazio início
 - 2.1.1 calcula similaridade s considerando o campo efeito dos objetos da lista de meta análogos
 - 2.1.2 calcula diferença d entre m usos possíveis que descrevem o objeto problema e a similaridade s
 - 2.1.3 $r = 1$
 - 2.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 2.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 fim_enquanto
3. se tipo improvisação igual a objetivo sem efeito conhecido então
 - 3.1 enquanto lista de objetos meta-meio análogos é diferente de vazio início
 - 3.1.1 calcula similaridade s considerando características particulares de tipo, área e subárea dos objetos da lista de meta-meio análogos e usos possíveis
 - 3.1.2 calcula diferença d entre m características particulares que descrevem o objeto problema e a similaridade s
 - 3.1.3 $r = 2$
 - 3.1.4 calcula a probabilidade $p(d,r) = N_m / N$
 - 3.1.5 armazena relacionamento entre objeto análogo e probabilidade de similaridade na lista de analogia
 fim_enquanto
4. seleciona da lista de analogia o objeto que possui maior probabilidade de similaridade
5. envia tipo de improvisação e objeto selecionado para submódulo da transformação da analogia em intenção

Figura 11.23: Algoritmo de alto nível da construção da analogia para a arquitetura de guias virtuais

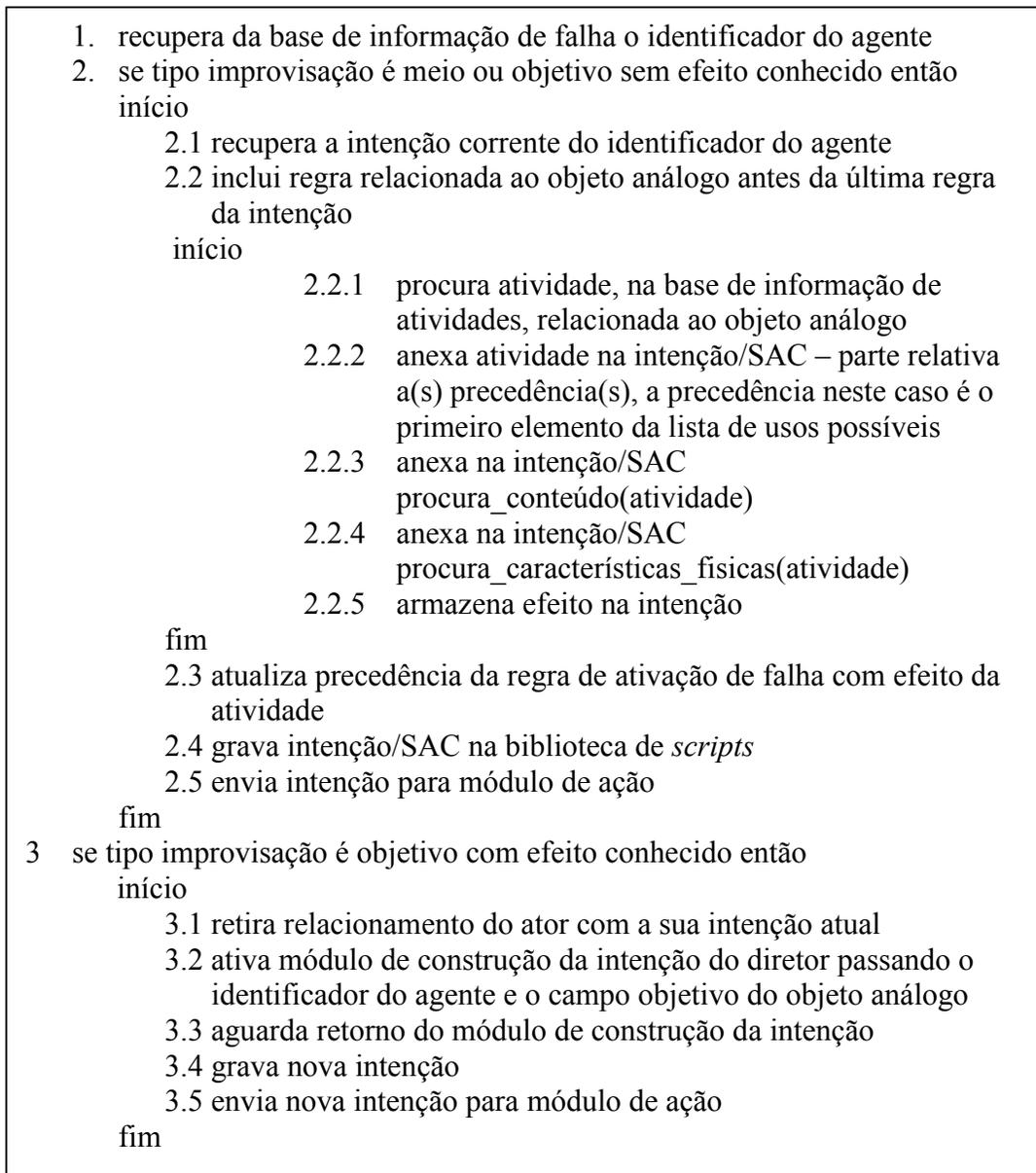


Figura 11.24: Algoritmo de alto nível da transformação da analogia em intenção para a arquitetura de guias virtuais

Quando a intenção é atualizada, a sua precedência é o primeiro elemento da lista de usos possíveis, pois este elemento foi o que ocasionou a falha durante a execução do *script* pelo ator. Desta maneira, quando o ator receber a atualização da intenção ele executa a regra relacionada ao objeto que gerou a falha do *script*.

11.5.2 Funcionamento do Módulo para o Papel de Ator

No papel de ator, o módulo não possui a mesma arquitetura que a apresentada na figura 11.21, pois no papel de ator o agente não é responsável por gerar uma solução para o problema que ocorreu. O ator é responsável por informar ao diretor que uma falha ocorreu e

requisitar uma solução para a mesma. Desta maneira, a arquitetura do módulo para o papel de ator possui duas funções básicas: uma responsável por obter maiores informações sobre o objeto que ocasionou a falha e outra para solicitar ao diretor uma solução para a falha.

O módulo é ativado no papel do ator pelo módulo de construção da intenção do ator no momento que ocorre uma falha. A falha durante uma apresentação de conteúdos pode ser a solicitação de um conteúdo que não está presente na intenção do agente. O módulo recebe como entrada o conteúdo solicitado que gerou a falha. A primeira coisa que o módulo deve fazer é obter informações adicionais sobre o conteúdo. As informações que ele irá solicitar estão relacionadas à estrutura da base de conteúdo, como o tipo do conteúdo, sua área e subárea. Tendo estas informações, o módulo ativa o módulo de ação para enviar uma mensagem ao diretor informando que ocorreu uma falha e passando a descrição do objeto que gerou a falha e a indicação do agente relacionado à falha.

O funcionamento do módulo para o papel de ator é bem simples se comparado ao funcionamento do módulo para o papel de diretor, pois no papel de ator o seu único propósito é obter maiores informações sobre a falha e requisitar uma solução para o diretor. Desta maneira, a improvisação para solução de problemas por analogia é executada por um agente que atua como diretor na arquitetura de guias virtuais.

11.5.3 Funcionamento do Módulo para o Papel de Diretor-Ator

No papel de diretor-ator o agente irá executar as duas versões do módulo de improvisação para solução de problemas. Se estiver executando o papel de diretor, ele possui a arquitetura apresentada na figura 11.21 e executa os algoritmos descritos nas figuras 11.22, 11.23 e 11.24. No caso contrário, executa as duas funções descritas para o funcionamento do módulo no papel de ator.

11.6 Módulo de Percepção

Este módulo será sensível a diferentes percepções dependendo do papel que o agente estiver executando. Como mencionado anteriormente, e apresentado na tabela 11.4, a estrutura de uma percepção é composta por tipo, identificador do agente, e descrição de um objeto. O tipo de percepção irá variar dependendo do papel que o agente está executando.

Se o agente está executando o papel de diretor ele pode receber percepções do autor de uma apresentação e dos seus agentes atores. O diretor irá receber percepções do autor no momento em que estiver executando o seu módulo de aquisição de conhecimento e irá receber percepção dos atores no momento em que eles estiverem executando uma intenção como um SAC. As percepções que o diretor pode receber dos atores são, inicialmente, de dois tipos: indicação de final de intenção e indicação de falha na intenção. Se a percepção indica *fim* de intenção, e está sendo realizada uma direção descentralizada, ou seja, o agente executa o papel de diretor-ator, o diretor finaliza a sua própria intenção. Se a percepção indica *fim*, e está sendo realizada uma direção centralizada, o diretor retira o relacionamento do ator com a sua intenção atual. Se não existe mais nenhum relacionamento ativo, então significa que todos os atores finalizaram as suas apresentações e o diretor pode finalizar a sua própria intenção. No caso da percepção indicar *falha* de intenção, o módulo de percepção grava na base de informação de falha o identificador do

agente que ocasionou a falha e envia para o módulo de improvisação para solução de problemas a descrição do objeto que ocasionou a falha.

Se o agente estiver executando o papel de ator, o módulo irá receber percepções do ambiente externo. Estas percepções podem indicar, inicialmente, dois tipos de requisições para o agente ator: a execução da próxima atividade da intenção ou a requisição por um assunto específico.

Se esta percepção for um indicativo de próxima atividade, o módulo de percepção recupera o efeito da lista de efeitos gerados pela execução das atividades da intenção, relaciona o efeito à percepção e envia para o escalonador de atividades da construção da intenção do ator.

Se esta percepção for diferente de um indicativo de próxima atividade, o módulo de percepção envia a percepção como recebida para o escalonador de atividades da construção da intenção do ator.

11.7 Módulo de Ação

O módulo de ação recebe requisições do módulo de construção da intenção e do módulo de improvisação para solução de problemas. Contudo, estas requisições dependem do tipo de papel que o agente está executando.

Se o agente está executando o papel de diretor, ele recebe do módulo de construção da intenção uma intenção para enviar a um determinado agente ator. Ele também pode receber do módulo de improvisação para solução de problemas uma intenção para enviar a um determinado agente ator.

Se o agente está executando o papel de ator, ele recebe da construção de intenção do ator uma requisição de execução do conteúdo e característica física selecionada para performance. O módulo também pode receber uma requisição da improvisação de solução de problemas do ator para enviar o pedido de solução de uma falha ao diretor. Neste caso, o módulo de ação deve enviar para a o módulo de percepção do diretor o identificador do agente que ocasionou a falha e a descrição do objeto problema.

12 ESTUDO DE CASO: FUNCIONAMENTO DA ARQUITETURA DE AGENTES IMPROVISACIONAIS PARA O MUSEU VIRTUAL SAGRES

O SAGRES [BER 99] é um museu virtual destinado a apoiar a aprendizagem, explorando aspectos relevantes para a interação entre o visitante e os recursos do museu, e, também, entre um conjunto de visitantes.

O SAGRES foi construído com base na estrutura física do Museu de Ciências e Tecnologia (MCT) da Pontifícia Universidade Católica do Rio Grande do Sul. Os assuntos de consulta estão dispostos no sistema de acordo com sua localização dentro da área de exposições do MCT, permitindo ao visitante encontrar os experimentos desejados no caso de uma visita real. A organização de uma base de informações, muitas não presentes na área de exposições, também foi prevista, já que um dos objetivos era fazer com que o visitante também utilizasse o SAGRES como uma fonte de consulta pós-visita, podendo adquirir informações complementares às encontradas no museu físico. O visitante pode, inclusive, encontrar no SAGRES informações sobre as pesquisas desenvolvidas nos laboratórios do MCT, bem como sobre o acervo [MOR 2003a].

Através desta base de informações, o SAGRES proporciona todo o conhecimento do museu ao alcance da comunidade, uma vez que os visitantes possuem acesso às informações do museu físico em uma exibição *on-line*.

As próximas seções apresentam o funcionamento dos módulos da arquitetura de guias virtuais para o museu virtual SAGRES, considerando um roteiro de apresentação sobre o conteúdo dinossauros.

12.1 Módulo de Aquisição de Conhecimento

Como mencionado no capítulo 11 seção 11.3, as estruturas da base de atividade e da base de conteúdo podem ser estendidas para incluir as características particulares do domínio que se deseja modelar. Desta maneira, a estrutura da base de conteúdo foi expandida a fim de representar como as informações estão organizadas no museu virtual SAGRES. Além das informações sobre tipo de conhecimento, área, subárea e textos, a base de informações deve guardar as características de local e pavimento (tabela 12.1). Local informa se as informações estão relacionadas a um laboratório do museu ou a área de exposição, e pavimento informa qual pavimento da área de exposições possui o experimento relacionado a um determinado conteúdo. Esta extensão é válida para o tipo de

conhecimento específico, para o conhecimento geral se mantém a estrutura padrão apresentada na tabela 11.3.

Tabela 12.1: Estrutura da base de informação do conteúdo para o SAGRES

Usos Possíveis	Características particulares				
	Tipo	Área	Subárea	Local	Pavimento

Nesta seção iremos descrever as trocas de mensagens realizadas entre um agente diretor e um escritor para aquisição de um roteiro para uma visita sobre dinossauros. Também mostraremos como estas informações seriam armazenadas dentro das bases de informação.

Ao iniciar uma troca de mensagem, o agente diretor solicita o objetivo da visita e a atividade inicial da mesma. Ao receber uma resposta do escritor, o agente gera um identificador para o objetivo e armazena as informações na base de informações de curso de ação. No caso da visita sobre dinossauros, a informação seria armazenada como mostra a tabela 12.2. O campo efeito somente será preenchido ao final da descrição do curso de ação.

Tabela 12.2: Estrutura de armazenamento do curso de ação para uma apresentação sobre dinossauros

Código	Objetivo	Efeito	Atividade Inicial
0001	dinossauros		Apresentação do guia

A partir deste momento, o agente diretor passa a requisitar as atividades que fazem parte deste curso de ação. A primeira atividade a ser descrita é a atividade inicial, neste caso a apresentação do guia. Serão requisitados o nome, a precedência, o efeito, as características físicas e os usos possíveis da atividade. Para atividade *Apresentação do guia*, a informação seria apresentada como mostra a tabela 12.3.

Tabela 12.3: Estrutura de armazenamento da atividade apresentação do guia

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação pessoal Saudação	Apresentação do guia	nenhuma	Apresentação do guia OK	Abanar Falar Explicar

O campo efeito sempre possui um efeito gerado pelo agente diretor indicando que a atividade foi realizada com sucesso, o que irá possibilitar a construção da intenção no módulo de construção do SAC/Intenção. Além disso, o campo efeito possui uma indicação da consequência da realização da atividade.

Após receber as informações sobre as atividades, o agente diretor solicita ao escritor as informações sobre o conteúdo relacionado a atividade *Apresentação do Guia* (tabela 12.4).

Tabela 12.4: Estrutura de armazenamento do conteúdo apresentação do guia

Usos Possíveis	Características particulares			
	Tipo	Área	Subárea	Textos
Apresentação pessoal Saudação Identificação pessoal	Geral	Pessoal	Identificação	Olá!... Oi!... Bem vindo... Seja bem vindo...

Enquanto o escritor não indica que a atividade descrita é a última do curso de ação, o agente diretor continua solicitando as atividades e os conteúdos relacionados a mesma. Uma seqüência de atividades e conteúdos possíveis seria a apresentada abaixo.

Tabela 12.5: Estrutura de armazenamento da atividade antes dos dinossauros

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Antes dos dinossauros	Apresentação do guia	Antes dinossauros OK	Falar Explicar

Tabela 12.6: Estrutura de armazenamento do conteúdo antes dos dinossauros

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Antes dos dinossauros Pré-história Dinossauros Evolução da terra	Específico	História	Milhões de Anos	Exposição	2 ^o Pavimento	Antes... Os dinossauros... Viveram...

Tabela 12.7: Estrutura de armazenamento da atividade vestígios de dinossauros

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Vestígios de dinossauros	Antes do dinossauro	Vestígios dinossauros OK	Falar Explicar

Tabela 12.8: Estrutura de armazenamento do conteúdo vestígios de dinossauros

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Vestígios de dinossauros Pré-história Dinossauros Evolução da terra Répteis primatas	Específico	História	Milhões de Anos	Exposição	2 ^o Pavimento	Existem... Vestígios... Nos dias atuais... Os vestígios...

Tabela 12.9: Estrutura de armazenamento da atividade um novo dinossauro em Santa Maria

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Um novo dinossauro em Santa Maria	Antes do dinossauro	Novo dinossauro OK	Falar Explicar

Tabela 12.10: Estrutura de armazenamento do conteúdo um novo dinossauro em Santa Maria

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Dinossauro em Santa Maria Descoberta Arqueológica Pré-história Dinossauros Evolução da terra Santa Maria	Específico	História	Milhões de Anos	Exposição	2 ^o Pavimento	Em Santa Maria... Os dinossauros... Foram encontrados... Encontraram em...

Tabela 12.11: Estrutura de armazenamento da atividade o passado

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	O passado	Antes do dinossauro	Passado OK	Falar Explicar

Tabela 12.12: Estrutura de armazenamento do conteúdo o passado

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Passado Dinossauros Evolução da Terra	Específico	História	Milhões de Anos	Exposição	2 ^o Pavimento	No passado... Antigamente...

Tabela 12.13: Estrutura de armazenamento da atividade o presente

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	O presente	O passado e um novo dinossauro em Santa Maria e Vestígios Dinossauros	Presente OK	Falar Explicar

Tabela 12.14: Estrutura de armazenamento do conteúdo o presente

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Presente Dinossauros Evolução da Terra	Específico	História	Milhões de Anos	Exposição	2 ^o Pavimento	Atualmente... No presente... Hoje...

Tabela 12.15: Estrutura de armazenamento da atividade finalizar apresentação

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Finalização Término da apresentação	Finalizar apresentação	O presente	Fim	Falar Explicar Finalizar Abanar

Tabela 12.16: Estrutura de armazenamento do conteúdo finalizar apresentação

Usos Possíveis	Características particulares			
	Tipo	Área	Subárea	Textos
Finalização Término da apresentação	Geral	Pessoal	Finalização	Até logo... Foi bom... Espero que... Volte... Até mais...

Como a atividade finalizar apresentação é a última atividade do curso de ação, ela possui como efeito um indicador para o final do curso de ação. Quando o agente diretor identifica a última atividade, ele envia uma última mensagem para o escritor solicitando o efeito do objetivo, ou seja, do curso de ação como um todo e atualiza a base de informação do curso de ação.

Tabela 12.17: Atualização da estrutura de armazenamento do curso de ação para uma apresentação sobre dinossauros

Código	Objetivo	Efeito	Atividade Inicial
0001	dinossauros	Pré-história	Apresentação do guia

12.1.1 Considerações quanto aos Tipos de Improvisação que Podem Ocorrer Durante esta Apresentação

Podemos observar que durante a aquisição de conhecimento, o escritor e o agente diretor podem deixar em aberto questões que possibilitam a improvisação do texto do agente. Como o escritor não indica uma seqüência completamente ordenada de apresentação, o agente poderá escolher qual ordem apresentar os conteúdos de vestígios de dinossauros, um novo dinossauro em Santa Maria e o passado. Além disso, o agente pode improvisar seus comportamentos físicos e verbais, pois as características físicas da atividade possuem mais de uma maneira de serem executadas e o mesmo acontece com os textos relacionados aos conteúdos. Desta maneira, o agente pode improvisar seu comportamento através do seu curso de ação e capacidades físicas e verbais.

12.2 Módulo de Construção da Intenção

Como mencionado no capítulo anterior, este módulo pode ser executado por um agente exercendo papel de diretor, ator ou diretor-ator e em cada um destes casos o módulo opera em um nível de improvisação do texto. Como diretor, o módulo opera no nível de preparação da improvisação do texto, construindo um *script* abstrato de comportamento que será destinado a um ator. Como ator, o módulo opera no nível de execução da improvisação de texto, construindo uma instânciação do *script* abstrato de comportamento para execução. Como diretor-ator ele opera nos dois níveis, dependendo do papel que estiver executando em um determinado momento.

Para um ator ter uma intenção é necessário que ele a receba de um diretor, seja este outro agente ou ele próprio exercendo o papel de diretor. Desta maneira, o módulo inicia executando um papel de diretor.

12.2.1 Execução para o Papel de Diretor

Após adquirir conhecimento no módulo aquisição de conhecimento, o diretor inicia o processo de construção do *script* relacionado ao conhecimento adquirido. Assim, ele executa o algoritmo descrito na figura 11.13 para construção da intenção como um script abstrato de comportamento. Considerando o conteúdo adquirido anteriormente, o diretor procura na base de informações de curso de ação por um objetivo que seja igual a *dinossauros*. Ao encontrar o objetivo, ele verifica se a intenção está vazia. Como a intenção está vazia ele procura pela atividade inicial cuja precedência seja nenhuma e encontra a atividade *Apresentação do guia*. Ao encontrar a atividade, ele inclui a atividade na lista de atividades da intenção, anexa a parte relativa à precedência na intenção, chama o escalonador de comportamento, armazena efeito na intenção e atualiza a variável de efeito com efeito atual, ou seja, *Apresentação do guia OK*. Após executar esta seqüência de passos, a intenção que está em construção possui a estrutura apresentada na figura 12.1.

<pre> 1. Se precedência = nenhum então início 1.1 procura_conteúdo(apresentação do guia) 1.2 procura_características_fisicas(apresentação do guia) 1.3 efeito = apresentação do guia OK fim </pre>
--

Figura 12.1: Inclusão da atividade inicial na intenção

Após incluir a atividade inicial, a execução do algoritmo prossegue, pois o efeito é diferente de fim, mas como a intenção não é mais vazia o algoritmo entra no senão e, enquanto existir atividade cujo efeito seja igual ao efeito atual, ele verifica se a atividade não existe na lista de atividades da intenção, se ela não existir ele armazena o efeito da atividade como efeito intermediário, anexa a parte relativa à precedência na intenção, chama o escalonador de comportamento, armazena efeito na intenção. Caso a atividade já exista na lista de atividades, ele procura pela próxima atividade que possua a precedência. A única atividade que possui precedência igual ao efeito *Apresentação do guia OK* é a atividade *Antes dos dinossauros*. Desta maneira, o efeito intermediário recebe o efeito *Antes dinossauros OK*, e a intenção é atualizada como mostra a figura 12.2.

- | |
|--|
| <ol style="list-style-type: none"> 1. Se precedência = nenhum então <ul style="list-style-type: none"> início 1.1 procura_conteúdo(apresentação do guia) 1.2 procura_características_físicas(apresentação do guia) 1.3 efeito = apresentação do guia OK fim 2. se precedência = apresentação do guia OK então <ul style="list-style-type: none"> início 2.1 procura_conteúdo(antes dos dinossauros) 2.2 procura_características_físicas(antes dos dinossauros) 2.3 efeito = antes dinossauros OK fim |
|--|

Figura 12.2: Inclusão da atividade antes dos dinossauros na intenção

Como não existem mais atividades relacionadas ao efeito *Apresentação do guia OK*, o efeito é atualizado com o efeito intermediário *Antes dinossauros OK* e o algoritmo prossegue sempre entrando no senão da intenção vazia. No exemplo, existem três atividades cuja precedência é igual a *Antes dinossauros OK*. Desta maneira, ao executar o enquanto existir atividade cujo efeito seja igual ao efeito atual, ele encontra a primeira atividade na base de informações de atividade que é *Vestígios de dinossauros*. Se a atividade não existe na lista de atividades da intenção, então inclui atividade na lista de atividades da intenção, armazena efeito da atividade na lista de efeitos intermediários, anexa a parte relativa à precedência na intenção, chama o escalonador de comportamento, e procura pela próxima atividade com precedência igual a *Antes dinossauros OK*. Encontra a atividade *Um novo dinossauro em Santa Maria* e repete o procedimento descrito acima. Procura pela próxima atividade com precedência igual a *Antes dinossauros OK*. Encontra a atividade *O passado*, realiza a mesma seqüência de passos, sempre incluindo o efeito na lista de efeitos intermediários e procura pela próxima atividade com precedência igual a *Antes dinossauros OK*. Neste momento, a intenção já possui mais três novas atividades como mostra a figura 12.3.

Como não existem mais precedências iguais a *Antes dinossauros OK*, o algoritmo pega o primeiro efeito da lista de efeitos intermediários, no caso *Vestígios dinossauros OK*, e procura por precedências iguais a este efeito. É encontrada a atividade *O presente* cuja precedência contém o efeito como uma conjunção com outros dois efeitos, *O passado e um novo dinossauro em Santa Maria*. Se atividade não existe na lista de atividades da intenção, então insere atividade na lista de atividades da intenção, armazena efeito da atividade na lista de efeitos intermediários, anexa a parte relativa à precedência na intenção, chama o escalonador de comportamento e procura pela próxima atividade com precedência igual a *Vestígios dinossauros OK*. Como não existem mais atividades com precedência igual a *Vestígios dinossauros OK*, pega próximo efeito da lista de efeitos intermediários, o efeito passa a ser *novo dinossauro em Santa Maria*, procura por atividade cuja precedência seja igual a este efeito, vai encontrar a atividade *O presente*. Contudo, esta atividade já existe na lista de atividades, logo procura por outra atividade, como não existe, pega o próximo efeito da lista de efeitos que é *passado OK*. Procura por atividade que tenha esta precedência, recupera a atividade *O presente*, como esta atividade já existe na lista de

atividades, procura por outra atividade. Não existem mais atividades para o *passado OK*. Desta maneira, quando uma atividade possui precedências compostas por vários efeitos, ela somente é armazenada uma vez na intenção, como mostra a figura 12.4.

Pega próximo efeito na lista de efeitos intermediários, o *presente OK*. Procura por atividade que possua precedência igual ao efeito, encontra a atividade *Finalizar apresentação*. Como a atividade não existe na lista de atividades da intenção, então armazena a intenção e grava o efeito fim na lista de efeitos intermediários. Como não existem mais atividades para o efeito *presente OK*, atualiza o efeito com o efeito intermediário fim. Como o efeito é igual a fim sai do laço de enquanto que inclui as atividades na intenção.

Ao sair do laço da inclusão das atividades inclui atividade relativa à identificação de falha na intenção. Através da lista de atividades da intenção, constrói uma precedência indicando se a precedência é diferente de todas as precedências anteriores, então ocorreu uma falha e envia uma mensagem para o diretor requisitando recuperação da falha e passa o estado da intenção para falha. A figura 12.4 mostra a intenção completa para o objetivo de apresentação de conteúdos sobre *dinossauros*.

Logo após, o diretor relaciona um ator à intenção e grava a intenção como *script* abstrato de comportamento na biblioteca de *scripts*.

1. Se precedência = nenhum então
 - início
 - 1.1 procura_conteúdo(apresentação do guia)
 - 1.2 procura_características_físicas(apresentação do guia)
 - 1.3 efeito = apresentação do guia OK
 - fim
2. se precedência = apresentação do guia OK então
 - início
 - 2.1 procura_conteúdo(antes dos dinossauros)
 - 2.2 procura_características_físicas(antes dos dinossauros)
 - 2.3 efeito = antes dinossauros OK
 - fim
3. se precedência = antes dinossauros OK então
 - início
 - 3.1 procura_conteúdo(vestígios de dinossauros)
 - 3.2 procura_características_físicas(vestígios de dinossauros)
 - 3.3. efeito = vestígios dinossauros OK
 - fim
4. se precedência = antes dinossauros OK então
 - início
 - 4.1 procura_conteúdo(um novo dinossauro em Santa Maria)
 - 4.2 procura_características_físicas(um novo dinossauro em Santa Maria)
 - 4.3 efeito = novo dinossauro Santa Maria OK
 - fim
5. se precedência = antes dinossauros OK então
 - início
 - 5.1 procura_conteúdo(o passado)
 - 5.2 procura_características_físicas(o passado)
 - 5.3 efeito = passado OK
 - fim

Figura 12.3: Inclusão das atividades vestígios de dinossauros, um novo dinossauro em Santa Maria e o passado na intenção

1. Se precedência = nenhum então
 - início
 - 1.1 procura_conteúdo(apresentação do guia)
 - 1.2 procura_características_físicas(apresentação do guia)
 - 1.3 efeito = apresentação do guia OK
 - fim
2. se precedência = apresentação do guia OK então
 - início
 - 2.1 procura_conteúdo(antes dos dinossauros)
 - 2.2 procura_características_físicas(antes dos dinossauros)
 - 2.3 efeito = antes dinossauros OK
 - fim
3. se precedência = antes dinossauros OK então
 - início
 - 3.1 procura_conteúdo(vestígios de dinossauros)
 - 3.2 procura_características_físicas(vestígios de dinossauros)
 - 3.3. efeito = vestígios dinossauros OK
 - fim
4. se precedência = antes dinossauros OK então
 - início
 - 4.1 procura_conteúdo(um novo dinossauro em Santa Maria)
 - 4.2 procura_características_físicas(um novo dinossauro em Santa Maria)
 - 4.3 efeito = novo dinossauro Santa Maria OK
 - fim
5. se precedência = antes dinossauros OK então
 - início
 - 5.1 procura_conteúdo(o passado)
 - 5.2 procura_características_físicas(o passado)
 - 5.3 efeito = passado OK
 - fim
6. se precedência = vestígios dinossauros OK e novo dinossauro Santa Maria OK e passado OK então
 - início
 - 6.1 procura_conteúdo(presente)
 - 6.2 procura_características_físicas(presente)
 - 6.3 efeito = presente OK
 - fim
7. se precedência = presente OK então
 - início
 - 7.1 procura_conteúdo(finalizar apresentação)
 - 7.1 procura_características_físicas(finalizar apresentação)
 - 7.3 efeito = fim
 - fim
8. se precedência diferente de apresentação do guia OK e antes dinossauro OK e vestígios dinossauros OK e novo dinossauro Santa Maria OK e passado OK e presente OK então envia_mensagem_falha_diretor(precedência)

Figura 12.4: Estrutura Final da intenção para objetivo dinossauros

12.2.2 Execução para o Papel de Ator

Ao receber uma mensagem indicando que está relacionado a uma intenção como um *script* abstrato de comportamento, o ator ativa seu módulo de construção da execução da intenção, mais especificamente o submódulo do escalonador de atividades. A partir deste momento, o escalonador de atividades executa conforme o algoritmo de alto nível descrito na figura 11.19. O algoritmo recupera a intenção relacionada ao agente e enquanto a percepção é diferente de fim, ele seleciona a atividade cuja precedência é satisfeita. No início da execução não existe nenhuma precedência entre as atividades, desta maneira, o escalonador escolhe a primeira atividade. Como existe uma única atividade relacionada à precedência nenhum, ele armazena a atividade na lista de atividades executadas, chama o escalonador de comportamento para procurar por conteúdo para atividade apresentação do guia.

O escalonador de comportamento executa conforme o algoritmo de alto nível descrito na figura 11.20. O escalonador de comportamento verifica o tipo de mensagem, se é requisição de conteúdo ou requisição de características particulares e executa a seqüência de passos apropriada. Como a requisição é por conteúdo, ele verifica se não existe a lista de conteúdo para esta atividade, como não existe ele recupera a lista de conteúdos da base de informação de conteúdos. A lista resultante desta pesquisa é: *Olá!..., Oi!..., Bem vindo..., Seja bem vindo...* Logo após, o algoritmo seleciona randomicamente um texto da lista, retira da lista o texto selecionado e envia o texto para o escalonador de atividades.

O escalonador de atividades armazena o texto a ser apresentado e continua a sua execução chamando novamente o escalonador de comportamento, agora para pesquisar as características físicas da atividade. O escalonador de comportamento identifica que é requisição por características físicas, verifica se já possui lista de características físicas para atividade, como não possui, recupera a lista de características físicas da base de informação de atividades. A lista resultante desta pesquisa é: *abandar, falar e explicar*. Logo após, seleciona randomicamente uma característica da lista, retira a características selecionada da lista e envia a característica selecionada para o escalonador de atividades.

O escalonador de atividades envia o conteúdo e a característica selecionada para o módulo de ação responsável por executar os comportamentos do agente. Depois gera o efeito da atividade, no caso, *Apresentação guia OK*, e o armazena na lista de efeitos gerados. O último efeito armazenado é o efeito atual.

Após o agente executar o conteúdo e característica selecionada, ele fica aguardando uma nova percepção. Quando uma nova percepção acontece, o módulo de percepção capta a mesma e identifica o seu tipo. Se esta percepção for um indicativo de próxima atividade, o módulo de percepção recupera o efeito da lista de efeitos gerados, relaciona o efeito à percepção e envia para o escalonador de atividades. O escalonador de atividades recupera a atividade cuja precedência seja igual a *Apresentação guia OK*, que é a atividade *Antes dos dinossauros*. Como a precedência não é composta e a atividade não existe na lista de atividades executadas, o escalonador de atividades executa o mesmo processo descrito para a atividade anterior, gravando a atividade na lista de atividades executadas, armazenando o efeito *Antes dinossauros OK* na lista de efeitos gerados, enviando o conteúdo e a característica selecionada para o módulo de ação e esperando por uma nova percepção.

Ao chegar uma nova percepção, o módulo de percepção identifica que é um indicativo de próxima atividade e recupera o efeito da lista de efeitos gerados. O efeito recuperado é *Antes dinossauros OK*. A percepção envia o efeito para o escalonador de atividades. O escalonador de atividades identifica que existem três atividades relacionadas a precedência *Antes dinossauros OK*, que são *Vestígios de dinossauros*, *Um novo dinossauro em Santa Maria* e *O passado*. O escalonador de atividades ativa a função para escolha da atividade. Esta função segue o mesmo padrão da função para escolha de conteúdo e características físicas. Ela escolhe aleatoriamente entre a lista de atividades possíveis, ao escolher retira da lista a atividade escolhida e grava novamente a percepção na lista de efeitos gerados. Como ainda existem atividades relacionadas a esta percepção o efeito permanece na lista de efeitos gerados. Após ter selecionado a atividade, o algoritmo executa os mesmos passos realizados para as atividades anteriores, ativando o escalonador de comportamento, enviando os comportamentos selecionados para o módulo de ação e gerando o efeito da atividade.

Ao receber uma nova percepção, o módulo de percepção recupera o primeiro efeito da lista de efeitos gerados, que ainda é *Antes dinossauros OK*, e passa para o escalonador de atividades. O escalonador de atividades escolhe entre as atividades restantes da lista, e o processo segue como descrito no parágrafo acima. Desta maneira, enquanto existirem atividades relacionadas à precedência *Antes dinossauros OK*, o escalonador de comportamento executa a atividade e armazena o efeito da mesma no final da lista de efeitos gerados. Ao ser escolhida a última atividade da precedência *Antes dinossauros OK*, o efeito é retirado da lista de efeitos gerados, restando somente os efeitos das atividades executadas através da ativação da precedência *Antes dinossauros OK*.

Presumindo que a ordem de escolha das atividades fosse, *O passado*, depois *Vestígios de dinossauros* e por último *Um dinossauro em Santa Maria*, a lista de efeitos gerados iria ser composta, nesta ordem, pelos seguintes efeitos: *passado OK*, *vestígios dinossauros OK* e *dinossauro Santa Maria OK*. Desta maneira, a próxima percepção iria recuperar *o passado OK* e enviar para o escalonador de atividades. Quando o escalonador de atividade fosse recuperar a atividade relacionada a esta precedência, *o presente*, ele descobre que existem outras condições que fazem parte da precedência, no caso, *novo dinossauro em Santa Maria OK* e *vestígios dinossauros OK*. O escalonador descobre que a precedência é composta e ativa a função de solução de precedência composta.

A função de solução de precedência composta verifica se as demais precedências existem na lista de efeitos gerados. Se elas existirem, verifica, para cada uma das precedências, se elas possuem outras atividades relacionadas, caso elas não possuam mais atividades, retira as precedências da lista de efeitos gerados, no caso contrário os efeitos permanecem na lista de efeitos gerados.

Como todas as precedências estão na lista de efeitos gerados e nenhuma delas possui outras atividades relacionadas, além da atividade atual, elas são retiradas da lista de efeitos gerados que passa a ser vazia. Neste momento, o escalonador de atividades segue o seu fluxo normal, chamando o escalonador de comportamento, enviando o conteúdo e a característica física para o módulo de ação e armazenando o efeito *presente OK* na lista de efeitos gerados.

A próxima percepção irá recuperar o efeito *presente OK* como a precedência atual, e o escalonador de atividade encontra a atividade *Finalizar Apresentação* e segue seu fluxo normal, armazenando o efeito *fim* na lista de efeitos gerados.

A próxima percepção irá recuperar o efeito *fim* e como a percepção não será diferente de *fim*, o algoritmo do escalonador de atividades é finalizado e a apresentação de conteúdos termina.

A simulação apresentada acima descreve o funcionamento do algoritmo para uma execução sem falhas. Uma falha ocorre quando o módulo de percepção recebe uma percepção diferente da indicação de uma próxima atividade, o que ocasiona a impossibilidade de encontrar a percepção como um efeito na lista de efeitos gerados. Quando isto ocorre o módulo de percepção envia para o escalonador de atividade a percepção recebida. No escalonador de atividade a percepção irá ativar a mensagem *envia_mensagem_falha_diretor(precedência)* na intenção como SAC e o escalonador de atividade ativa o módulo de improvisação para solução de problemas para se comunicar com o diretor.

12.3 Módulo de Improvisação de Solução de Problemas

Assim como nos módulos apresentados anteriormente, o módulo de improvisação de solução de problemas irá agir de acordo com o papel que o agente está executando. Nas próximas seções será apresentado o funcionamento para o papel de ator e para o papel de diretor. O funcionamento para o papel de diretor-ator é a união do funcionamento do diretor e do ator. Neste módulo, a descrição do funcionamento do papel do ator é realizada antes do diretor, pois o ator é o responsável por identificar que ocorreu uma falha e por solicitar a solução deste problema ao diretor.

12.3.1 Funcionamento para o Papel de Ator

Quando o agente está executando o papel de ator, o módulo de improvisação para solução de problemas recebe do módulo de construção da intenção uma requisição de falha. Ao receber esta requisição, o módulo deve solicitar ao ambiente externo maiores informações sobre o conteúdo que gerou a falha. Considerando as bases de informações relacionadas ao domínio do SAGRES, o módulo pode solicitar informações adicionais sobre o tipo do conteúdo, área, subárea, local, pavimento e assuntos que estariam relacionados ao conteúdo. Com as informações adicionais obtidas, o módulo constrói o formato da requisição para o diretor, que é composto por tipo de requisição, identificador do agente que ocasionou a falha e descrição do objeto problema. A descrição do objeto problema é composta por duas estruturas: usos possíveis e características particulares. A descrição do objeto é montada da seguinte maneira: no campo usos possíveis são colocados o conteúdo que gerou a falha e os assuntos que estariam relacionados ao conteúdo; nas características particulares seriam colocados os campos relativos ao tipo, área, subárea, local e pavimento.

Uma falha ocorre quando uma informação solicitada não está presente na intenção como SAC atual do agente. Como mencionado nos capítulos 10 e 11, mais especificamente nas seções 10.3.2 e 11.5, as falhas podem ocasionar dois tipos de improvisação para solução de problemas: improvisação de meio e improvisação de objetivo.

Considerando a apresentação sobre dinossauros que está relacionada à área de história e à subárea milhões de anos, uma falha que ocasionaria uma improvisação de meio poderia ser a solicitação de um conteúdo sobre dinossauros que não foi descrito na intenção atual como SAC e que está relacionado à história e à subárea no passado por exemplo. Uma falha de improvisação de objetivo poderia ser a solicitação de um conteúdo diferente de dinossauro, ou a solicitação para fazer uma visita a algum laboratório do museu por exemplo.

Para ilustrar o funcionamento da improvisação de meio, suponha que o conteúdo solicitado ao agente ator tenha sido sítio arqueológico. Este conteúdo irá gerar uma falha no módulo de construção da intenção, pois não existe precedência relacionada a este conteúdo. Desta maneira, o módulo de improvisação de solução de problemas iria solicitar informações adicionais sobre este conteúdo. As informações que ele iria receber seriam as mostradas na tabela 12.18. A tabela mostra que o tipo de conteúdo e o pavimento não foram informados.

Tabela 12.18: Informações adicionais sobre conteúdo sítio arqueológico

Usos Possíveis	Características particulares				
	Tipo	Área	Subárea	Local	Pavimento
Dinossauros Arqueologia	----	História	No passado	Exposição	----

Com base nas informações da tabela 12.18, o módulo constrói uma percepção para ser enviada ao diretor. Considerando que o agente ator número 002 foi quem gerou a falha, a percepção gerada possuiria a estrutura ilustrada na tabela 12.19.

Tabela 12.19: Formato da percepção de falha para o conteúdo sítio arqueológico

Tipo	Identificador do Agente	Descrição do Objeto					
		Usos possíveis	Características particulares				
			tipo	área	subárea	local	pavimento
Falha	AT002	Sítio arqueológico Dinossauros Arqueologia	----	História	No passado	Exposição	----

Para ilustrar o funcionamento da improvisação de objetivo com efeito conhecido, suponha que o conteúdo solicitado ao agente ator tenha sido evolução humana e dos primatas. Este conteúdo irá gerar uma falha no módulo de construção da intenção, pois não existe precedência relacionada a este conteúdo. Desta maneira, o módulo de improvisação de solução de problemas iria solicitar informações adicionais sobre este conteúdo. As informações que ele iria receber seriam as mostradas na tabela 12.20. A tabela mostra que o tipo de conteúdo e o pavimento não foram informados.

Tabela 12.20: Informações adicionais sobre conteúdo evolução humana e dos primatas

Usos Possíveis	Características particulares				
	Tipo	Área	Subárea	Local	Pavimento
Evolução da terra Pré-história	----	História	No passado	Exposição	----

Com base nas informações da tabela 12.20, o módulo constrói uma percepção para ser enviada ao diretor. Considerando que o agente ator número 002 foi quem gerou a falha, a percepção gerada possuiria a estrutura ilustrada na tabela 12.21.

Tabela 12.21: Formato da percepção de falha para o conteúdo evolução humana e dos primatas

Tipo	Identificador do Agente	Descrição do Objeto					
		Usos possíveis	Características particulares				
			tipo	área	subárea	local	pavimento
Falha	AT002	Evolução humana e dos primatas Evolução da Terra Pré-história	----	História	No passado	Exposição	----

Para ilustrar o funcionamento da improvisação de objetivo sem efeito conhecido, suponha que o conteúdo solicitado ao agente ator tenha sido meteoritos. Este conteúdo irá gerar uma falha no módulo de construção da intenção, pois não existe precedência relacionada a este conteúdo. Desta maneira, o módulo de improvisação de solução de problemas iria solicitar informações adicionais sobre este conteúdo. As informações que ele iria receber seriam as mostradas na tabela 12.22. A tabela mostra que o tipo de conteúdo não foi informado.

Tabela 12.22: Informações adicionais sobre conteúdo evolução humana e dos primatas

Usos Possíveis	Características particulares				
	Tipo	Área	Subárea	Local	Pavimento
Planetas	----	História	Planeta Terra	Exposição	2 ⁰ Pavimento

Com base nas informações da tabela 12.22, o módulo constrói uma percepção para ser enviada ao diretor. Considerando que o agente ator número 002 foi quem gerou a falha a percepção gerada possuiria a estrutura ilustrada na tabela 12.23.

Tabela 12.23: Formato da percepção de falha para o conteúdo meteoritos

Tipo	Identificador do Agente	Descrição do Objeto					
		Usos possíveis	Características particulares				
			tipo	área	subárea	local	pavimento
Falha	AT002	Meteoritos Planetas	----	História	Planeta Terra	Exposição	2 ^o Pavimento

12.3.2 Funcionamento para o Papel de Diretor

Esta seção descreve o funcionamento do módulo de improvisação para solução de problemas do diretor para a improvisação de meio, improvisação de objetivo com efeito conhecido e improvisação de objetivo sem efeito conhecido.

12.3.2.1 Improvisação de Meio

Ao receber uma descrição de objeto problema do módulo de percepção, o módulo de improvisação para solução de problemas do diretor começa a executar o seu submódulo de identificação do tipo de improvisação. Suponha que o módulo tenha recebido a descrição apresentada na tabela 12.24.

Tabela 12.24: Descrição do objeto problema para sítios arqueológicos

Usos possíveis	Descrição do Objeto				
	Características particulares				
	tipo	área	subárea	local	pavimento
Sítio arqueológico Dinossauros Arqueologia	----	História	No passado	Exposição	----

O submódulo executa o algoritmo descrito na figura 11.22 e procura na base de informação de curso de ação por um objeto com objetivo compatível com um dos usos possíveis do objeto problema. O algoritmo encontra o objetivo dinossauros compatível com o uso possível do objeto problema. Como encontrou um objetivo compatível, o tipo de improvisação é de meio. Neste caso o algoritmo gera uma lista de meio análogos possíveis, procurando na base de informação de conteúdos por objetos que tenham características particulares compatíveis com a característica do objetivo problema e usos possíveis compatíveis com usos possíveis do objeto problema.

Considerando que além dos objetos descritos no módulo de aquisição de conhecimento, a base de informação de atividade e de conteúdo possuem os objetos descritos nas tabelas 12.25 e 12.26, a lista de objetos meio análogos será composta pelos objetos descritos nas tabelas 12.6, 12.8, 12.10, 12.12, 12.14 e 12.26. Após construir a lista, o submódulo envia o tipo de improvisação, a lista de objetos meio análogos e a descrição do objeto problema para o submódulo de construção da analogia.

Tabela 12.25: Estrutura de armazenamento da atividade sítio arqueológico

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Sítio arqueológico	nenhuma	Sítio arqueológico OK	Falar Explicar

Tabela 12.26: Estrutura de armazenamento do conteúdo sítio arqueológico

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Sítio arqueológico	Específico	História	No passado	Exposição	2 ⁰ Pavimento	Os sítios... A arqueologia... Existem...

O submódulo construção da analogia recebe a sua entrada e inicia o processamento descrito na figura 11.23. Ele verifica o tipo de improvisação, como é igual a meio ele percorre a lista de objeto meio análogos, calculando a similaridade, diferença e probabilidade para cada um dos objetos análogos possíveis. O algoritmo calcula a similaridade baseada nos usos possíveis e nas características particulares tipo, área, subárea, local e pavimento. Para o domínio de apresentação do SAGRES foram consideradas duas características adicionais em relação às características propostas na arquitetura geral de guias virtuais. A similaridade para os objetos 14, 16, 18, 20 e 22 é 3, pois somente as características área e local são similares às características do objeto problema e o uso possível dinossauro se encontra em todos os objetos. A diferença para estes objetos é 2, pois o número de atributos que descrevem o objeto problema são 5 (todas as características particulares). Considerando o número de atributos relevantes como 2 a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \frac{\binom{m-d}{r}}{\binom{m}{r}} = \frac{\binom{5-2}{2}}{\binom{5}{2}} = \frac{3!}{2!(3-2)!} \bigg/ \frac{5!}{2!(5-2)!} = 0,3$$

A similaridade para o objeto 34 é 4. A diferença é 1. Considerando o número de atributos relevantes como 2, a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \frac{\binom{m-d}{r}}{\binom{m}{r}} = \frac{\binom{5-1}{2}}{\binom{5}{2}} = \frac{4!}{2!(4-2)!} \bigg/ \frac{5!}{2!(5-2)!} = 0,6$$

O algoritmo armazena na lista de analogia o relacionamento entre o objeto análogo e a similaridade. Após examinar todos os objetos meio possíveis da lista, o algoritmo escolhe da lista de analogia o que possui maior similaridade. O objeto escolhido foi o objeto 34. São enviados para o submódulo transformação da intenção em analogia, o tipo de improvisação e o objeto meio selecionado.

O submódulo transformação da analogia em intenção executa conforme o algoritmo descrito na figura 11.24. O submódulo recupera da base de informação de falha o identificador do agente que gerou a falha, no caso o AT 002. Como o tipo de improvisação é igual a meio, ele recupera a intenção atual do agente AT002 e inclui a regra relacionada ao objeto análogo antes da última regra da intenção. Ele procura a atividade relacionada ao objeto análogo, que é a descrita na tabela 12.25, anexa na intenção a parte relativa à precedência, anexa na intenção a procura por conteúdo e características físicas da atividade, armazena o efeito na intenção, atualiza a precedência da regra de ativação da falha para conter o efeito da atividade atual, grava a intenção na biblioteca de *scripts* e ativa o módulo de ação. A figura 12.5 ilustra a intenção atualizada do agente AT002. Os passos 1, 2, 3, 4, 5, 6 e 7 são os mesmos apresentados na figura 12.4 e por este motivo não são descritos novamente. As atualizações são destacadas em negrito.

```

1. ....
...
7. ....
8. se precedência = sitio arqueológico então
  início
    8.1 procura_conteúdo(sitio arqueológico)
    8.2 procura_características_físicas(sítio arqueológico)
    8.3 efeito = sítio arqueológico OK
  fim
9. se precedência diferente de apresentação do guia OK e antes
dinossauro OK e vestígios dinossauros OK e novo dinossauro Santa
Maria OK e passado OK e presente OK e sitio arqueológico OK
então envia_mensagem_falha_diretor(precedência)

```

Figura 12.5: Atualização da intenção através da improvisação de meio

12.3.2.2 Improvisação de Objetivo sem Efeito Conhecido

Ao receber uma descrição de objeto problema, o módulo de improvisação para solução de problemas do diretor começa a executar o seu submódulo de identificação do tipo de improvisação. Suponha que o módulo tenha recebido a descrição apresentada na tabela 12.27.

Tabela 12.27: Descrição do objeto problema para meteoritos

Usos possíveis	Características particulares				
	tipo	área	subárea	local	pavimento
Meteoritos	----	História	Planeta	Exposição	2 ⁰
Planetas			Terra		Pavimento

O submódulo executa o algoritmo descrito na figura 11.22 e procura na base de informação de curso de ação por um objeto com objetivo compatível com um dos usos possíveis do objeto problema. Como não encontra nenhum, o algoritmo procura na base de informação de curso de ação por um objeto que possua efeito compatível com algum dos

usos possíveis do objeto problema. Se a lista de objetos resultante da procura é vazia, o submódulo gera uma lista de objetos meta-meio análogos procurando na base de informação de conteúdo por objetos com usos possíveis compatíveis com os usos possíveis e com características particulares compatíveis com características do objeto problema. Considerando que além dos objetos descritos no módulo de aquisição de conhecimento, a base de informação de atividade e de conteúdo possuem os objetos descritos nas tabelas 12.28 e 12.29, a lista de objetos meta-meio análogos será composta pelos objetos descritos nas tabelas 12.6, 12.8, 12.10, 12.12, 12.14 e 12.29. Após construir a lista, o submódulo identifica que o tipo de improvisação é de objetivo sem efeito conhecido e envia o tipo de improvisação, a lista de objetos meta-meio análogos e a descrição do objeto problema para o submódulo de construção da analogia.

Tabela 12.28: Estrutura de armazenamento da atividade minerais

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Minerais	Apresentação do guia	Minerais OK	Falar Explicar

Tabela 12.29: Estrutura de armazenamento do conteúdo minerais

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Minerais	Específico	História	Planeta Terra	Exposição	2 ⁰ Pavimento	Os minérias... Ao longo dos anos... Muitos minerais...

O submódulo construção da analogia recebe a sua entrada e inicia o processamento descrito na figura 11.23. Ele verifica o tipo de improvisação, como é igual a objetivo sem efeito conhecido, ele percorre a lista de objeto meta-meio análogos, calculando a similaridade, diferença e probabilidade para cada um dos objetos análogos possíveis. O algoritmo calcula a similaridade baseada nos usos possíveis e nas características particulares tipo, área, subárea, local e pavimento. A similaridade para os objetos 14, 16, 18, 20 e 22 é 3, pois somente as características área, local e pavimento são similares as características do objeto problema. A diferença para estes objetos é 2, pois o número de atributos que descrevem o objeto problema são 5 (todas as características particulares). Considerando o número de atributos relevantes como 2 a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \binom{m-d}{r} / \binom{m}{r} = \binom{5-2}{2} / \binom{5}{2} = \frac{3!}{2!(3-2)!} / \frac{5!}{2!(5-2)!} = 0,3$$

A similaridade para o objeto 37 é 4. A diferença é 1. Considerando o número de atributos relevantes como 2, a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \frac{\binom{m-d}{r}}{\binom{m}{r}} = \frac{\binom{5-1}{2}}{\binom{5}{2}} = \frac{4!}{2!(4-2)!} \bigg/ \frac{5!}{2!(5-2)!} = 0,6$$

O algoritmo armazena na lista de analogia o relacionamento entre o objeto análogo e a similaridade. Após examinar todos os objetos meio possíveis da lista, o algoritmo escolhe da lista de analogia o que possui maior similaridade. O objeto escolhido foi o objeto 37. São enviados para o submódulo transformação da intenção em analogia, o tipo de improvisação e o objeto meta-meio selecionado.

O submódulo transformação da analogia em intenção executa conforme o algoritmo descrito na figura 11.24. O submódulo recupera da base de informação de falha o identificador do agente que gerou a falha, no caso o AT002. Como o tipo de improvisação é igual a objetivo sem efeito conhecido, ele recupera a intenção atual do agente AT002 e inclui a regra relacionada ao objeto análogo antes da última regra da intenção. Ele procura a atividade relacionada ao objeto análogo, que é a descrita na tabela 12.28, anexa na intenção a parte relativa à precedência, anexa na intenção a procura por conteúdo e características físicas da atividade, armazena o efeito na intenção, atualiza a precedência da regra de ativação da falha para conter o efeito da atividade atual, grava a intenção na biblioteca de *scripts* e ativa o módulo de ação. A precedência neste caso é igual ao primeiro elemento da lista de usos possíveis do objeto problema.

A figura 12.6 ilustra a intenção atualizada do agente AT002. Os passos 1, 2, 3, 4, 5, 6 e 7 são os mesmos apresentados na figura 12.4 e por este motivo não são descritos novamente. As atualizações são destacadas em negrito.

```

1. ....
...
7. ....
8. se precedência = meteoritos então
início
8.1 procura_conteúdo(minerais)
8.2 procura_características_físicas(minerais)
8.3 efeito = meteoritos OK
fim
9. se precedência diferente de apresentação do guia OK e antes
dinossauro OK e vestígios dinossauros OK e novo dinossauro Santa
Maria OK e passado OK e presente OK e meteoritos OK então
envia_mensagem_falha_diretor(precedência)

```

Figura 12.6: Atualização da intenção através da improvisação de objetivo sem efeito conhecido

Pode-se observar na figura 12.6 que a analogia produzida é fraca no sentido de que não representa o conteúdo que foi solicitado, ela representa um conteúdo que se encontra

em uma mesma subárea, mas isto não garante que seja o próprio conteúdo. Neste caso, o algoritmo poderia ser aperfeiçoado para fazer com que o agente ator informasse que não foi possível encontrar o conteúdo exato, mas que encontrou outro conteúdo dentro da mesma área e subárea do solicitado.

12.3.2.3 Improvisação de Objetivo com Efeito Conhecido

Ao receber uma descrição de objeto problema, o módulo de improvisação para solução de problemas do diretor começa a executar o seu submódulo de identificação do tipo de improvisação. Suponha que o módulo tenha recebido a descrição apresentada na tabela 12.30.

Tabela 12.30: Descrição do objeto problema para evolução humana e dos primatas

Descrição do Objeto					
Usos possíveis	Características particulares				
	tipo	área	subárea	local	pavimento
Evolução humana e dos primatas Evolução da Terra Pré-história	----	História	No passado	Exposição	----

O submódulo executa o algoritmo descrito na figura 11.22 e procura na base de informação de curso de ação por um objeto com objetivo compatível com um dos usos possíveis do objeto problema. Como não encontra nenhum, o algoritmo procura na base de informação de curso de ação por objetos que possuam efeito compatível com algum dos usos possíveis do objeto problema, gerando uma lista de objetos meta análogos. Considerando que além dos objetos descritos anteriormente, as bases de informações possuem os objetos descritos nas tabelas 12.31, 12.32, 12.33, 12.34 e 12.35, a lista será composta de dois objetos meta análogos possíveis, que são os objetos apresentados nas tabelas 12.17 e 12.31. Como a lista não é vazia o tipo de improvisação é de objetivo com efeito conhecido. O submódulo envia o tipo de improvisação e a lista de objetos meta análogos possíveis para o submódulo de construção da analogia.

Tabela 12.31: Estrutura de armazenamento do curso de ação para uma apresentação sobre estágios da terra

Código	Objetivo	Efeito	Atividade Inicial
0002	Estágios da terra	Evolução da terra Evolução humana e dos primatas	Sítio Arqueológico

Tabela 12.32: Estrutura de armazenamento da atividade evolução humana e dos primatas

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Apresentação de conteúdo	Evolução humana e dos primatas	Sítio arqueológico OK	Evolução humana primatas OK	Falar Explicar

Tabela 12.33: Estrutura de armazenamento do conteúdo evolução humana e dos primatas

Usos Possíveis	Características particulares					
	Tipo	Área	Subárea	Local	Pavimento	Textos
Evolução humana e dos primatas	Específico	História	No passado	Exposição	2 ^o Pavimento	A evolução... Antes de... A terra...

Tabela 12.34: Estrutura de armazenamento da atividade finalizar apresentação para curso de ação estágios da terra

Usos Possíveis	Características particulares			
	Nome	Precedência	Efeito	Físicas
Finalização	Finalizar apresentação	Evolução humana primatas OK	Fim	Falar Finalizar Abanar

Tabela 12.35: Estrutura de armazenamento do conteúdo finalizar apresentação para curso de ação estágios da terra

Usos Possíveis	Características particulares			
	Tipo	Área	Subárea	Textos
Finalização Término da apresentação	Geral	Pessoal	Finalização	Até logo... Foi bom... Espero que... Volte... Até mais...

O submódulo construção da analogia recebe sua entrada e processa o algoritmo descrito na figura 11.23. O tipo de improvisação é objetivo com efeito conhecido, desta maneira, enquanto a lista de objetos meta análogos é diferente de vazia, calcula a similaridade, diferença e probabilidade de similaridade para cada uma das opções. A similaridade s é calculada considerando o atributo m como usos possíveis e comparando estes com o campo efeito do objeto meta análogo. O objeto 25 possui similaridade 1, pois somente um dos usos possíveis do objeto problema se encontra na lista de efeitos deste objeto. Sendo m igual a 3, pois são três os usos possíveis do objeto problema, a diferença d

é 2. Sendo 1 o número de atributos relevantes, e a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \binom{m-d}{r} / \binom{m}{r} = \binom{3-2}{1} / \binom{3}{1} = \frac{1!}{1!(0)!} / \frac{3!}{1!(3-1)!} = 0,33$$

O objeto 39 possui similaridade 2, pois dois dos usos possíveis do objeto problema estão na lista de efeitos do objeto 39. A diferença é 1, o número de atributos relevantes é 1 e a probabilidade de similaridade $p(d,r) = N_m / N$ é calculada como:

$$= \binom{m-d}{r} / \binom{m}{r} = \binom{3-1}{1} / \binom{3}{1} = \frac{2!}{1!(2-1)!} / \frac{3!}{1!(3-1)!} = 0,66$$

O algoritmo armazena na lista de analogia o relacionamento entre o objeto análogo e a similaridade. Após examinar todos os objetos meio possíveis da lista, o algoritmo escolhe da lista de analogia o que possui maior similaridade. O objeto escolhido foi o objeto 39. São enviados para o submódulo transformação da intenção em analogia, o tipo de improvisação e o objeto meta selecionado.

O submódulo transformação da analogia em intenção executa conforme o algoritmo descrito na figura 11.24. O submódulo recupera da base de informação de falha o identificador do agente que gerou a falha, no caso o AT002. Como o tipo de improvisação é igual a objetivo com efeito conhecido, ele retira o relacionamento do ator com a sua intenção atual. Logo após, ele ativa o módulo de construção da intenção passando o identificador do agente e o objetivo do objeto meta selecionado e fica aguardando um retorno do módulo de construção da intenção. Ao receber a nova intenção do módulo de construção da intenção, ele grava a intenção e envia para o módulo de ação a nova intenção e o identificador do agente para o qual esta intenção deve ser submetida.

12.4 Considerações sobre o Capítulo

Neste capítulo foi apresentado o funcionamento de cada um dos módulos da arquitetura de guias virtuais para o domínio do museu virtual SAGRES. Foram apresentadas as extensões realizadas nas características particulares dos objetos para que eles refletissem a estrutura da base de informações do SAGRES. Desta maneira, além do tipo, área e subárea e textos, foram incluídas as características particulares relacionadas à localização do conteúdo, em relação à área de exposição ou laboratório, e o pavimento físico do museu onde este conteúdo pode ser encontrado.

Os cursos de ação, atividades e conteúdos descritos fazem parte da base de informações do museu virtual SAGRES e foram estruturados a fim de ilustrar o funcionamento dos algoritmos descritos no capítulo 11. Com estas informações foi possível realizar uma simulação dos módulos de aquisição de conhecimento, construção da intenção e improvisação de solução de problemas para cada um dos possíveis papéis dos agentes.

Futuramente, as informações que constituirão as bases de informação dos cursos de ação, das atividades e dos conteúdos serão obtidas através do grupo do museu responsável pela construção de roteiros e visitas guiadas ao SAGRES.

Com a simulação dos módulos se pode observar como ocorrem os dois tipos de improvisação disponibilizados pela arquitetura: a improvisação de textos e a improvisação de solução de problemas. A improvisação de texto, conhecida no teatro improvisacional como improvisação implícita, ocorre através do módulo de construção da intenção como SAC e é desenvolvida através de satisfação de restrições. A improvisação de solução de problemas ocorre no módulo com o mesmo nome e é desenvolvida como uma analogia por similaridade.

Foram ilustrados os três tipos de improvisação possíveis por analogia: de meio, de objetivo sem efeito conhecido e de objetivo com efeito conhecido. Destas analogias, a de objetivos sem efeito conhecido foi identificada como a mais fraca, pois está relacionada ao caso em que o agente não encontra um objetivo ou efeito de objetivo compatível com um objeto problema. Contudo, para não ocorrer um erro como não encontrar um objeto compatível o agente procura por algum objeto que tenha características particulares semelhantes. Isto nem sempre é o mais adequado, pois o objeto encontrado pode estar na mesma área, mas não na mesma subárea do objeto problema, por exemplo. Este é o caso do agente explicar que não conseguiu achar um conteúdo que satisfizesse uma requisição, mas que ele encontrou outros conteúdos que podem ter relação com o que foi solicitado.

Neste sentido, através desta simulação foi possível se verificar a aplicabilidade da arquitetura de guias virtuais e a utilidade da improvisação, tanto para a construção de intenções como *scripts* abstratos de comportamento quanto para a solução de problemas que podem ocorrer durante uma execução.

13 CONCLUSÕES

Este trabalho defende a tese de que o uso da improvisação em agentes improvisacionais, vistos como um tipo de agente deliberativo, possibilita que os agentes improvisem comportamentos e solucionem problemas de maneira coerente, criando e mantendo uma ilusão de vida para o agente e contribuindo para o aperfeiçoamento dos sistemas multiagentes.

Para comprovar nossa tese, foram necessários vários esforços. O primeiro deles foi o de inserir as questões de improvisação como aspectos importantes dentro da área de Inteligência Artificial, mas especificamente de sistemas multiagentes. Neste sentido, iniciamos o trabalho buscando identificar a utilização de conceitos de improvisação dentro de agentes de interface responsáveis por interagir com o usuário e a importância destes aspectos para construção de um agente credível [MOR 99b] [MOR 99c] [COS 2000]. Concluímos em [MOR 2001a] [MOR 2001b] que a improvisação está presente implicitamente em agentes representados através de personagens animados e figuras humanas. Contudo, sentimos que isto não era suficiente, era necessário mostrar uma ligação mais forte entre a improvisação e a Inteligência Artificial. Lançamos um novo esforço no sentido de examinar as diferenças e semelhanças entre improvisação e planejamento. Após alguns estudos contatamos em [MOR 2002a] [MOR 2002b] que a improvisação considera a idéia de planos como intenções, ao contrário do planejamento clássico que considera planos como programas [AGR 88a]. Com base nisto surgiu a idéia de entender agentes improvisacionais como um tipo de agente deliberativo e não como uma nova categoria de agentes como proposto por Hayes-Roth [HAY 98b]. Iniciamos um novo esforço no sentido confirmar nossa hipótese.

A fim de mostrar que agentes improvisacionais são um tipo de agentes deliberativos, escolhemos analisar uma arquitetura específica de agentes deliberativos e identificar a existência de aspectos de improvisação na mesma. Optamos pela arquitetura BDI por ser uma arquitetura tradicional na área. Realizamos estudos sobre a origem desta arquitetura e sobre algumas arquiteturas BDI particulares.

Com base nas fundamentações teóricas do teatro improvisacional, identificamos quatro tipos de improvisação possíveis e analisamos as arquiteturas BDI estudadas a fim de mostrar a existência de improvisação nas mesmas. Como discutido no capítulo 9, as arquiteturas BDI possuem improvisação de texto, que está relacionada à improvisação na construção do curso de ação dos agentes. Isto demonstrou que os agentes deliberativos

possuíam características de improvisação. Contudo, estas características estavam relacionadas a um tipo de improvisação considerada como implícita. Além da improvisação implícita, os agentes improvisacionais devem ter capacidade de agir frente a algo inesperado, que tenha acontecido por acaso e para o qual não estavam preparados, eles deve executar improvisação explícita. Esta característica estava faltando nos agentes deliberativos a fim de que se tornassem verdadeiros agentes improvisacionais. Desta maneira, expandimos a arquitetura BDI para uma arquitetura BDI improvisacional, que possui um módulo responsável pela improvisação da solução de problemas. Esta improvisação é realizada através de analogia por similaridade, como apresentado no capítulo 10. De acordo com Russell [RUS 88] a analogia por similaridade é uma técnica apropriada para ambientes complexos que exijam respostas rápidas. Desta maneira, esta técnica é bastante apropriada para a improvisação explícita.

Outro esforço desta tese estava em examinar o uso de um agente diretor no processo de coordenação de agentes atores. A figura do diretor não tem sido considerada nos trabalhos anteriores de improvisação. A ênfase destes trabalhos está sempre nos processos dos atores, sendo que o diretor é um agente humano responsável por informar as instruções aos atores, como apresentado no capítulo 3. Desta maneira, foram estudados os aspectos relativos à direção improvisacional (capítulo 5) e à coordenação como um processo de direção improvisacional (capítulo 7). Este estudo gerou um novo espaço de coordenação para sistemas multiagentes que considera as questões de improvisação e distribuição de direção [MOR 2003b] [MOR 2003c] [MOR 2003d].

Outro esforço foi lançado no sentido de unir uma arquitetura BDI improvisacional e os aspectos da direção improvisacional, o que resultou na arquitetura de guias virtuais apresentada no capítulo 11. A fim de mostrar a funcionalidade desta arquitetura e a maneira como ocorre a improvisação de texto, a improvisação de solução de problemas e a coordenação improvisacional, apresentamos a aplicação desta arquitetura para o museu virtual SAGRES (capítulo 12).

Concluimos que ao longo deste documento construímos os argumentos necessários para defesa da nossa tese e desenvolvemos as contribuições esperadas para a mesma. Na medida em que íamos desenvolvendo nossas contribuições, novas idéias surgiram e como não foi possível incorporá-las a tese neste momento, elas serão trabalhadas futuramente. Algumas destas idéias são apresentadas na próxima seção, onde são descritos os trabalhos futuros.

13.1 Trabalhos Futuros

Um primeiro trabalho futuro será a implementação dos algoritmos descritos na arquitetura de guias virtuais para o museu virtual SAGRES, a fim de validar a eficiência dos mesmos para os diferentes papéis que os agentes podem assumir na sociedade.

Pretendemos trabalhar em maiores detalhes o espaço de coordenação improvisacional, mais especificamente os eixos das técnicas e da distribuição de coordenação. Tendo realizado isto, podemos analisar a questão de uma coordenação que se

encontra entre os extremos da centralização e descentralização, para posteriormente incluir esta possibilidade de coordenação dentro da arquitetura de guias virtuais.

Outro trabalho futuro será o aperfeiçoamento da função de escolha das atividades e conteúdos do escalonador de comportamentos. Esta função pode ser aperfeiçoada através da inclusão de aspectos de personalidade dos agentes. Desta maneira, pretendemos examinar mais a fundo as questões relativas às personalidade e emoções dos agentes e seus relacionamento com as ações que expressam seus comportamentos.

Tendo realizado os estudos sobre personalidade, pretendemos desenvolver uma ferramenta que possibilite a construção de agentes improvisacionais, através da descrição de suas personalidades, comportamentos e ações.

Como as tarefas de um diretor improvisacional, descritos no capítulo 5, são a construção de *scripts* para atores, a avaliação e a resolução de problemas para coordenação e execução de uma apresentação e a nossa arquitetura, até o momento, examinou duas destas tarefas, pretendemos trabalhar o aspecto da avaliação tanto da performance dos atores quanto do próprio processo de direção improvisacional.

Também pretendemos realizar um estudo sobre a improvisação em agentes conversacionais, analisando o entendimento e a geração da conversação através de técnicas de improvisação.

Apesar de termos aplicado a arquitetura BDI improvisacional para um domínio de apresentação de conteúdos, acreditamos que esta arquitetura possa ser aplicada a outros domínios como por exemplo, animação de objetos em computação gráfica e futebol de robôs. Pretendemos examinar a aplicabilidade dos conceitos de improvisação e a arquitetura BDI improvisacional para outros domínios.

REFERÊNCIAS

- [AAM 94] AAMODT, A.; PLAZA, E. Case-Base Reasoning: Foundational Issues, Methodological Variations, and System Approaches. **AICom – Artificial Intelligence Communications**, Amsterdam, v. 7, n. 1, p. 39-59, 1994.
- [AGR 87] AGRE, P. E.; CHAPMAN, D. Pengi: An Implementation of a Theory of Activity. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI, 6., 1987. **Proceedings...** Los Altos, CA: Morgan Kaufmann, 1987. v. 1, p. 268-272.
- [AGR 88a] AGRE, P. E.; CHAPMAN, D. **What are plans for?** Cambridge: MIT Press, 1988. (MIT A.I. Memo 1050).
- [AGR 88b] AGRE, P. E. **Dynamic Structure of Everyday Life**. 1988. Tese (Doutorado em Ciência da Computação) – MIT Artificial Intelligence Laboratory, Cambridge, (Technical Report, 1085).
- [AGR 97] AGRE, P. E. **Computation and Human Experience**. [S.l.]: Cambridge University Press, 1997.
- [AGU 2003] AGUIRRE, J. O. O. **Lecture 2: Coordination Models**. Notas de aula. Disponível em : <<http://delta.cs.cinvestav.mx/~oolmedo/Coordination/2.pdf>>. Acesso em: ago. 2003.
- [AND 95] ANDERSON, J. E. **Constraint Directed Improvisation for Everyday Activities**. 1995. Tese (Doutorado em Ciência da Computação) - University of Manitoba, Manitoba.
- [BAT 91a] BATES, J. Virtual Reality, Art, and Entertainment. **Presence – The Journal of Teleoperators and Virtual Environments**, [S. l.], v.2, n.1, 1991.
- [BAT 91b] BATES, J; LOYALL, A.B.; REILLY, W.S. Broad Agents. In: AAAI SPRING SYMPOSIUM ON INTEGRATED INTELLIGENT ARCHITECTURES. AAAI, 1991, Stanford. **Proceedings...** Disponível em: <<http://www.cs.cmu.edu/Groups/oz/papers.html>>. Acesso em: mar.

- 2001.
- [BAT 92a] BATES, J; LOYALL, A.B.; REILLY, W.S. **An Architecture for Action, Emotion, and Social Behavior**. Pittsburgh: Carnegie Mellon University, 1992. (Technical Report, CMU-CS-92-144).
- [BAT 92b] BATES, J. **The Nature of Characters in Interactive Worlds and The OZ Project**. Pittsburgh: Carnegie Mellon University, 1992. (Technical Report, CMU-CS-92-200).
- [BAT 94] BATES, J. **The Role of Emotion in Believable Agents**. Pittsburgh: Carnegie Mellon University, 1994. (Technical Report, CMU-CS-94-136).
- [BEE 98] BEER, M. et al. Negotiation in Multi-Agent System. In: WORKSHOP OF THE UK SPECIAL GROUP ON MULTI-AGENT SYSTEMS, UKMAS, 1998. **Proceedings...** [S.l.:s.n.], 1998.
- [BER 2002] BERGENTI, F.; RICCI, A. Three Approaches to the Coordination of Multiagent Systems. In: SYMPOSIUM ON APPLIED COMPUTING, 2002. **Proceedings...** New York, NY: ACM Press, 2002. p. 367-372.
- [BER 98] BERGER, L; BERGER, S. **The Theatre Team: Playwright, Producer, Director, Designers, and Actors**. Westport, Connecticut: Greenwood Press, 1998.
- [BER 99] BERTOLETTI, A. C.; COSTA, A. C. R. SAGRES – A Virtual Museum. In: MUSEUMS AND THE WEB, 1999. **Proceedings...** New Orleans, Louisiana: [s.n.], 1999.
- [BLA 91] BLACK, George. **Contemporary Stage Directing**. Holt, Rinehart and Winston Inc., Chicago, 1991.
- [BLU 95] BLUMBERG, B.; GALYEAN, T. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments. **Computer Graphics**, Los Alamitos, v. 30, n. 3, p. 47-54, 1995.
- [BON 2002] BONARINI, A.; RESTELLI, M. **An Architecture for Adaptive Coordination**. Disponível em: <<http://www-dii.ing.unisi.it/aiia2002/paper/ROBOTICA/bonarini-aiia02.pdf>>. Acesso em: jul. 2002.
- [BRA 88] BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. Plans and Resource-Bounded Practical Reasoning. **Computational Intelligence**, Cambridge, v.4, n.4, p.349-355, 1988.
- [BRA 99a] BRATMAN, M. E. **Intention, Plans and Practical Reasoning**. Palo Alto: Leland Stanford Junior University: CSLI Publications, 1999.
- [BRA 99b] BRATMAN, M. E. **Faces of Intention: Selected Essays on Intention and Agency**. [S.l.]: Cambridge University Press, 1999.

- [BRO 76] BROCKETT, O. G. **The Essential Theater**. New York: Holt, Rinehart and Winston, 1976.
- [BRO 99] BROGI, A.; JACQUET, J-M. On The Expressiveness of Coordination Models. In: COORDINATION LANGUAGES AND MODELS, Coordination, 1999. **Proceedings...** [S.l.]: Springer, 1999. (Lecture Notes in Computer Science, 1594).
- [BUS 2001] BUSI, N.; CIANCARINI, P.; GORRIERI, R.; ZAVATTARO, G. Models for coordinating agents: a guided tour. In: COORDINATION MODELS AND APPLICATIONS FOR AGENTS, 2001. **Proceedings...** [S.l.]:Springer, 2001.
- [CAR 83] CARBONELL, J. G. Learning by analogy: Formulating and generalizing plans from past experience. In: MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, M. T. (Ed.). **Machine Learning: An Artificial Intelligence Approach**. Palo Alto:Tioga, 1983. p.137 – 162.
- [CAR 86] CARBONELL, J. G. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In: MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, M. T. (Ed.). **Machine Learning: An Artificial Intelligence Approach**. Los Altos: Morgan Kaufmann, 1986. p. 371 - 392.
- [CAR 99] CARDOSO, H. L. **Sistema Multi-Agente para Comércio Eletrônico**. 1999. Dissertação (Mestrado em Ciência da Computação) – Universidade do Porto, Porto. Disponível em: <<http://www.fe.up.pt/~eol/PROJECTS/SMACE/relatorio.html>>. Acesso em: jul. 2002.
- [CAR 2000] CARDOSO, H. L.; OLIVEIRA, E. Using and Evaluating Adaptive Agents for Electronic Commerce Negotiation. In: INTERNATIONAL JOINT CONFERENCE ADVANCES IN ARTIFICIAL INTELLIGENCE, IBERAMIA'2000/ SBIA'2000. **Proceedings...** [S.l.]:Springer, 2000. p. 96-105. (Lecture Notes in Artificial Intelligence, 1952).
- [CAS 99] CASTELFRANCHI, C. et al. Deliberate Normative Agents: Principles and Architecture. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 6., 1999. **Proceedings...** Orlando,FL: Springer,1999.
- [CAV 95a] CAVEDON, L. et al. Revisiting rationality for agents with intentions. In: AUSTRALIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 8., 1995. **Bringing the Gap: proceedings**. [S.l.]:World Scientific, 1995. p. 131-138.
- [CAV 95b] CAVEDON, L.; TIDHAR; G.; MORLEY, D. **A Framework for Modelling Multi-Agent Systems**. Melbourne: Australian Artificial Intelligence Institute, 1995. (Technical Report, 65).

- [CHA 83] CHACRA, Sandra. **Natureza e Sentido da Improvisação Teatral**. São Paulo: Perspectiva, 1983. 118p.
- [CHA 2001] CHALMERS, S.; GRAY, P. M. D. BDI agents and constraint logic. **AISB Journal Special Issue on Agent Technology**, [S.l.], v.1, n.1, p.21-40, 2001.
- [CHA 96] CHAVES, A.; MAES, P. Kasbah: An Agent Marketplace for Buying and Selling Goods. In: INTERNATIONAL CONFERENCE ON THE PRACTICAL APPLICATION OF INTELLIGENT AGENTS AND MULTI-AGENT TECHNOLOGY, PAAM, 1., 1996. **Proceedings...** [S.l.:s.n.], 1996.
- [CHE 86] CHEKHOV, M. **Para o ator**. São Paulo: Martins Fontes, 1986.
- [CIA 2000] CIANCARINI, P.; OMICINI, A.; ZAMBONELLI, F. Multiagent System Engineering: the Coordination Viewpoint. In: INT. WORKSHOP ON AGENT THEORIES, ARCHITECTURES AND LANGUAGES, 6., 2000. **Proceedings...** [S.l.:s.n.], 2000.
- [CIA 2001] CIANCARINI, P. **Coordination Languages and Models**. Notas de aula. Disponível em: <www.cs.unibo.it/~cianca/wwwpages/seminari/>. Acesso em: mar. 2002.
- [COE 74] COHEN, Robert; HARROP, John. **Creative Play Direction**. Englewood Cliffs , New Jersey: Prentice-Hall, 1974.
- [COS 2000] COSTA, A. C. R.; MORAES, M. C.; RODRIGUES, M. Improvisational Interface Agents for E-Commerce Systems. In: INTERNATIONAL ICSC SYMPOSIUM ON MULTI-AGENTS AND MOBILE AGENTS IN VIRTUAL ORGANIZATIONS AND E-COMMERCE. **Proceedings...** [S.l.:s.n.], 2000.
- [DAV 87] DAVIES, T. R; RUSSELL, S. A Logical Approach to Reasoning by Analogy. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 10., 1987. **Proceedings...** Milan, Italy: Morgan Kaufmann, 1987.
- [DAV 2001] DAVIES, J; GOEL, Ashok K. Visual Analogy in Problem Solving. In: JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, **Proceedings...** [S.l.:s.n.], 2001. p. 377-384.
- [DAR 83] DAVIS, R.; SMITH, R. G. Negotiation as a metaphor for distributed problem solving. **Artificial Intelligence**, [S.l.], v. 20, n. 1, p. 63-109. Jan. 1983.
- [DAR 88] DAVIS, R.; SMITH, R. G. Negotiation as a metaphor for distributed problem solving. In: BOND, A. H.; GASSER, L. (Ed.). **Readings in Distributed Artificial Intelligence**. Los Altos, San Mateo: Morgan

- Kaufmann, 1988.
- [DEC 2000] DECHTER, R.; ROSSI, F. Constraint Satisfaction. **Survey ECS**, [S.I.], Mar. 2000. Disponível em: <<http://www.ics.uci.edu/~dechter/publications/>>. Acesso em: jul. 2001.
- [DEN 98] DENTI, E.; NATALI, A.; OMICINI, A. On the expressive power of a language for programming coordination media. In: ACM SYMPOSIUM ON APPLIED COMPUTING, SAC, 1998. **Proceedings...** Atlanta: ACM Press, 1998. p. 169-177.
- [D'IN 97] D'INVERNO, M.; KINNY, D.; LUCK, M.; WOOLDRIDGE, M. **A Formal Specification of dMARS**. Melbourne: Australian Artificial Intelligence Institute, 1997. (Technical Report, 72).
- [DOR 96] DORAN, J. E.; FRANKLIN, S.; JENNINGS, N. R. On Cooperation in Multi-Agent Systems. In: UK WORKSHOP ON FOUNDATION OF MULTI-AGENT SYSTEMS, 1., 1996. **Proceedings...** [S.I.:s.n.], 1996.
- [DUR 89a] DURFEE, E.; LESSER, V.; CORKILL, D. Trends in cooperative distributed problem solving. **IEEE Trans. Knowl. Data Eng.**, New York, v.11, n.1, p. 63-83, 1989.
- [DUR 89b] DURFEE, E.; LESSER, V.; CORKILL, D. Cooperative distributed problem solving. In: BARR, A.; FELGENBAUM, E. A. (Ed.). **The Handbook of Artificial Intelligence**. [S.I.]:Addison Wesley, 1989. v. 4, p. 83-147.
- [DUR 99] DURFEE, E. Distributed Problem Solving and Planning. In: WEISS, G. (Ed.). **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence**. [S.I.]: The MIT Press, 1999. p. 121-164.
- [DUR 2001] DURFEE, E. Scaling Up Agent Coordination Strategies. **IEEE Computer**, New York, July 2001.
- [EME 89] EMERSON, E. A.; SRINIVASAN, J. Branching time temporal logic. In: LINEAR TIME, BRANCHING TIME AND PARTIAL ORDER IN LOGICS AND MODELS FOR CONCURRENCY, 1988. **Proceedings...** Berlin: Springer-Verlag, 1989. p. 123-172.
- [EXT 2000] EXTEMPO SYSTEMS INC. **User's Guide to the Imp Character Editor: Imp Character Development Kit v. 2.1**. Redwood City, CA, 2000.
- [FAR 98] FARATIN, P.; SIERRA, C.; JENNINGS, N. R. Negotiation Decision Functions for Autonomous Agents. **Int. Journal of Robotics and Autonomous Systems** **24**, Gibraltar, v. 3, n.4, p. 159-182, 1998.
- [FAR 99] FARATIN, P.; SIERRA, C.; JENNINGS, N. R.; BUCKLE, P. Designing Responsive and Deliberative Automated Negotiators. In: AAAI WORKSHOP ON NEGOTIATION: SETTLING CONFLICTS AND IDENTIFYING OPPORTUNITIES, 1999. **Proceedings...** Orlando,

- FL:AAAI Press, 1999. p. 12-18.
- [FER 99] FERREIRA, A. B. H. **Aurélio século XXI** : o dicionário da língua portuguesa. 3. ed. Rio de Janeiro: Nova Fronteira, 1999.
- [FRA 97] FRANKLIN, S.; GRAESSER, A. Is it an Agent or just a Program?: A Taxonomy for Autonomous Agents. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES,3., 1997. **Proceedings...** [S.l.]:Springer-Verlag 1997.
- [GEL 92] GELERNTER, D.; CARRIERO, N. Coordination Languages and Their Significance. **Communications of the ACM**, New York, v. 35, n. 2. Feb. 1992.
- [GEN 91] GENTNER, D.; FORBUS, K. D. MAC/FAC: A Model of Similarity-based Retrieval. In: COGNITIVE SCIENCE CONFERENCE, 13., 1991. **Proceedings...** Erlbaum, Hillsdale: [s.n.], 1991. p. 504-509.
- [GEO 89] GEORGEFF, M.; INGRAND, F. Decision-Making in an Embedded Reasoning System. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 11., 1989. **Proceedings...** [S.l.]: AAAI Press, 1989.
- [GEO 98] GEORGEFF, M.; PELL, B.; POLLACK, M.; WOOLDRIDGE, M. The Belief-Desire-Intention Model of Agency. In: INTERNATIONAL WORKSHOP ON INTELLIGENT AGENTS V: AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 5., 1998. **Proceedings...** [S.l.]:Springer-Verlag, 1998. p. 1-10.
- [GRE 93] GREINER, R. Adaptive Derivation Processes. In: WORKSHOP ON KNOWLEDGE COMPILATION AND SPEEDUP LEARNING, 1993. **Proceedings...** Amherst:[s.n.], 1993.
- [GUT 98a] GUTTMAN, R.; MAES, P. Agent-mediated Integrative Negotiation for Retail Electronic Commerce. In: WORKSHOP ON AGENT MEDIATED ELECTRONIC TRADING, AMET, 1998. **Proceedings...** [S.l.:s.n.], 1998.
- [GUT 98b] GUTTMAN, R.; MAES, P. Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In: INTERNATIONAL WORKSHOP ON COOPERATIVE INFORMATION AGENTS, 2., 1998. **Proceedings...** [S.l.:s.n.], 1998.
- [HAD 96] HADDADI, A. **Communication and Cooperation in Agent Systems: A Pragmatic Theory**. Berlin: Springer-Verlag, 1996. (Lecture Notes in Artificial Intelligence, 1056).
- [HAY 94a] HAYES-ROTH, B. et al. **Directed Improvisation**. Stanford: Stanford University, 1994. (Technical Report, KSL-94-61).
- [HAY 94b] HAYES-ROTH, B.; BROWNSTON, N.L. **Multi-Agent Collaboration in**

- Directed Improvisation.** Stanford: Stanford University, 1994. (Technical Report, KSL-94-69).
- [HAY 95a] HAYES-ROTH, Barbara. An Architecture for Adaptative Intelligent Systems. **Artificial Intelligence: Special Issue on Agents and Interactivity**, [S.l.], n. 72, p. 329-365, 1995.
- [HAY 95b] HAYES-ROTH, Barbara; BROWNSTON, Lee; SINCOFF, Erik. **Directed Improvisation by Computer Characters.** Stanford: Stanford University, 1995. (Technical Report, KSL-95-04).
- [HAY 95c] HAYES-ROTH, B. Directed Improvisation by Computer. In: STANFORD SEMINAR ON HUMAN-COMPUTER INTERACTION, 1995. Disponível em: <<http://www-pcd.stanford.edu/cs547>>. Acesso em: abr. 1997.
- [HAY 95d] HAYES-ROTH, B. et al. Directed Improvisation with Animated Puppets. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1995, Denver. **Proceedings...** Disponível em: <<http://ksl-web.stanford.edu/projects/cait/publicity.html>>. Acesso em: abr. 1999.
- [HAY 97a] HAYES-ROTH, Barbara; GENT, Robert van; HUBER, Daniel. Acting in Character. In: CREATING PERSONALITIES FOR SYNTHETIC ACTORS: TOWARDS AUTONOMOUS PERSONALITY AGENTS, 1997. **Proceedings...** Berlin: Springer-Verlag, 1997. p. 92-112.
- [HAY 97b] HAYES-ROTH, B.; VAN GENT, R. Story-Making with Improvisational Puppets. In: AGENTS, 1997, Marina Del Rey. **Proceedings...** Disponível em: <<http://sigart.acm.org/proceedings/agents97>>. Acesso em: jan. 2000.
- [HAY 98a] HAYES-ROTH, Barbara. **Interacting with Animated Characters:** Puppets, Bartenders, and Auto Salespersons. Stanford: Stanford University, 1998. (Technical Report, KSL-98-07).
- [HAY 98b] HAYES-ROTH, B.; DOYLE, P. Animate Characters. **Autonomous Agents and Multi-Agent Systems**, Amsterdam, v. 1, p. 195-230, 1998.
- [HAY 99a] HAYES-ROTH, B. Smart Interactive Characters: Automating One-To-One Customer Service. **Web Techniques**, San Francisco, Sept. 1999.
- [HAY 99b] HAYES-ROTH, B; GENT, R.V.; REYNOLDS, R.; JOHNSON, M.V.; WESCOURT, K. Web guides. **IEEE Intelligent Systems**, Los Alamitos, p. 23-27, Mar./Apr. 1999.
- [HAY 99c] HAYES-ROTH, B.; JOHNSON, V.; GENT, R.; WESCOURT, K. Intelligent agents in the form of personable characters interact with and serve human customers. **Communications of the ACM**, New York, v. 43, n. 3, p. 103-105, 1999.
- [HAY 2001] HAYES-ROTH, B. Adaptive Learning Guides. In: IASTED CONFERENCE ON COMPUTERS AND ADVANCED TECHNOLOGY

- IN EDUCATION, 2001. **Proceedings...** [S.l.:s.n.], 2001. Disponível em: <<http://www.extempo.com>>. Acesso em: jan. 2002.
- [HAL 89] HALL, R. Computational Approaches to Analogical Reasoning: A Comparative Analysis. **Artificial Intelligence**, [S.l.], v.39, p.39-120, 1989.
- [HUA 2002] HUANG, P.; SYCARA, K. A Computational Model for Online Agent Negotiation. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 35., 2002. **Proceedings...** [S.l.:s.n.], 2002.
- [HUH 99] HUHNS, M. N.; STEPHENS, L. M. Multiagent Systems and Societies of Agents. In: WEISS, G. (Ed.). **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence**. Cambridge: The MIT Press, 1999.
- [JEN 93a] JENNINGS, N. Specification and Implementation of a Belief-Desire-Joint-Intention Architecture for Collaborative Problem Solving. **Int. Journal of Intelligent and Cooperative Information Systems**, [S.l.], v.2, n.3, p. 289-318, 1993.
- [JEN 93b] JENNINGS, N. Controlling Cooperative Problem Solving Using Joint Intentions. **AI Communications**, Amsterdam, v.6, n.3&4, p. 247-248, 1993.
- [JEN 93c] JENNINGS, N. R. Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems. **The Knowledge Engineering Review**, Cambridge, v.8, n.3, p. 223-250, 1993.
- [JEN 96] JENNINGS, N. R. Coordination Techniques for Distributed Artificial Intelligence. In: O'HARE, G.M.P.; JENNINGS, N. R. (Ed.) **Foundations in Distributed Artificial Intelligence**. New York: Wiley, 1996. p.187-210.
- [JEN 2000] JENNINGS, N. R.; PARSONS, S.; SIERRA, C.; FARATIN, P. Automated Negotiation. In: INT. CONF. ON THE PRACTICAL APPLICATION OF INTELLIGENT AGENTS AND MULTI- AGENT SYSTEMS, PAAM, 5., 2000. **Proceedings...** Manchester, UK: [s.n.], 2000. p. 23-30.
- [JEN 2001] JENNINGS, N. et al. Automated negotiation: prospects, methods and challenges. **Int. J. of Group Decision and Negotiation**, Amsterdam, v.10, n.2, p. 199-215, 2001.
- [JEN 99] JENSEN, D. et al. Learning Quantitative Knowledge for Multiagent Coordination. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI, 16., 1999. **Proceedings...** [S.l.]: AAAI Press, 1999. p. 24-31.
- [KAN 90] KANTROWITZ, M. **Natural Language Text Generation in the Oz Interactive Fiction Project**. Pittsburg: Carnegie Mellon University, 1990. (Technical Report, CMU-CS-90-158).
- [KAN 92] KANTROWITZ, M. **Integrated Natural Language Generation Systems**.

- Pittsburg: Carnegie Mellon University, 1992. (Technical Report, CMU-CS-92-107).
- [KEL 91] KELSO, M.; WEYHRAUCH, P.; BATES, J. Dramatic Presence. **Presence – The Journal of Teleoperators and Virtual Environments**, [S.l.], v.2, n.1, 1991.
- [KIM 94] KIM, S. H. **Learning and Coordination**: enhancing agent performance through distributed decision making. Dordrecht : Kluwer Academic, 1994.
- [KIN 96] KINNY, D.; GEORGEFF, M.; RAO, A. **A Methodology and Modelling Technique for Systems of BDI agents**. Melbourne: Australian Artificial Intelligence Group, 1996. (Technical Report, 58).
- [KLU 2001] KLUSCH, M.; SYCARA, K. Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In: COORDINATION OF INTERNET AGENTS, 2001. **Proceedings...** [S.l.]:Springer, 2001.
- [KOD 96a] KODA, T.; MAES, P. **Agents with Faces**: The Effect of Personification. 1996. Disponível em: <<http://tomoko.www.media.mit.edu/groups/agents/publications/>>. Acesso em: jan. 2000.
- [KOD 96b] KODA, T. **Agents with Faces: A Study on the Effects of Personification of Software Agents**. 1996. Dissertação (Mestrado em Mídia e Ciências), MIT Program in Media Arts and Sciences. Disponível em: <<http://tomoko.www.media.mit.edu/people/tomoko/thesis.html>>. Acesso em: jan. 2000.
- [KON 93] KONOLIGE, K.; POLLACK, M. E. A Representationalist Theory of Intention. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, IJCAI, 13., 1993. **Proceedings...** San Mateo: Morgan Kaufmann, 1993. p. 390-395.
- [KRA 2001] KRAUS, Sarit. **Strategic negotiation in multiagent environments**. Cambridge: MIT, 2001.
- [KUM 92] KUMAR, V. Algorithms for Constraint Satisfaction Problems: A Survey. **AI Magazine**, Amsterdam, p. 32-44, Spring 1992.
- [LAU 93] LAUREL, Brenda. **Computers as Theatre**. San Francisco: Addison-Wesley. 1993.
- [LES 97a] LESTER, J. C. et al. The Persona Effect: Affective Impact of Animated Pedagogical Agents. In: CHI Electronic Publications, 1997. **Proceedings...** Disponível em: <<http://www.acm.org/sigchi/proceedings/paper/jl.htm>>. Acesso em: abr. 2000.
- [LES 97b] LESTER, J.C.; FITZGERALD, P. J.; STONE, B. The Pedagogical Design Studio: Exploiting Artifact-Based Task Models for Constructivist Learning. In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER

- INTERFACES, 3., 1997. **Proceedings...** [S.l.:s.n.], 1997. p.155-162.
- [LOY 91] LOYALL, A.B.; BATES, J. **Hap: A Reactive, Adaptive Architecture for Agents**. Pittsburgh: Carnegie Mellon University, 1991. (Technical Report, CMU-CS-91-147).
- [LOY 93] LOYALL, A.B.; BATES, J. Real-time Control of Animated Broad Agents. In: ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY, 15., 1993. **Proceedings...** Disponível em: <<http://www.cs.cmu.edu/Groups/oz/papers.html>>. Acesso em: ago. 2000.
- [LOY 97] LOYALL, B. **Believable Agents: Building Interactive Personalities**. 1997. Tese (Tese de Doutorado) – School of Computer Science, Carnegie Mellon University, Pittsburg, PA. (Technical Report, CMU-CS-97-123).
- [MAE 88] MAES, P.; NARDI, D. **Workshop Meta-Level Architectures and Reflection**. Amsterdam: North-Holland, 1988. 355 p.
- [MAL 94] MALONE, T. W.; CROWSTON, K. The Interdisciplinary Study of Coordination. **ACM Computing Surveys**, New York, v. 26, n. 2, p. 87-119, Mar. 1994.
- [MAR 98] MARRIOTT, K.; STUCKEY, P.J. **Programming with Constraints: An Introduction**. Cambridge:Massachusetts, MIT Press, 1998.
- [MAT 2002] MATEAS, M. **Interactive Drama, Art, and Artificial Intelligence**. 2002. Tese (Tese de Doutorado) – School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. (Technical Report, CMU-CS-02-206).
- [MCD 78] MCDERMOTT, J. **ANA: An assimilating and accommodating production system**. Pittsburgh: Carnegie-Mellon University, 1978. (Technical Report, CMU-CS-78-156).
- [MCD 79] MCDERMOTT, J. Learning to use analogies. In: IJCAI, 1979. **Proceedings...** Tokyo:[s.n.], 1979. p. 568-576.
- [MOR 99a] MORAES, M. C. **Uma arquitetura de agentes improvisacionais para apoio a visitação de museus baseada em computadores**. 1999. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, PUCRS, Porto Alegre.
- [MOR 99b] MORAES, M. C. COSTA, A.C. R. Uma arquitetura multi-agente para apoiar a visitação de museus virtuais. In: CONFERENCIA LATINOAMERICANA DE INFORMÁTICA, CLEI, 25., 1999. **Proceedings...** Assuncion, Paraguay:[s.n.], 1999. p. 737-746.
- [MOR 99c] MORAES, M. C.; COSTA, A. C. R. Virtual Guides to Assist Visitors in the SAGRES Virtual Museum. In: INTERNATIONAL CONFERENCE OF CHILEAN COMPUTER SCIENCE SOCIETY, 19., 1999. **Proceedings...** [S.l.:s.n.], 1999.

- [MOR 2001a] MORAES, M. C. **Agentes Improvisacionais: Uma Perspectiva Para Agentes de Software Voltados para Interfaces**. Exame de Qualificação (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [MOR 2001b] MORAES, M. C.; BERTOLETTI, A. C.; COSTA, A. C. R. Estudo e Avaliação da Usabilidade de Agentes Improvisacionais de Interface. In: WORKSHOP DE INTERFACES HOMEM-COMPUTADOR, 4., 2001. **Anais...** Florianópolis:UFSC:SBC, 2001.
- [MOR 2002a] MORAES, M. C. **Relatório das Atividades Desenvolvidas Durante Doutorado Sanduíche com Dra. Barbara Hayes-Roth**. 2002. Relatório Final de Bolsa de Doutorado Sanduíche – CAPES, Brasília.
- [MOR 2002b] MORAES, M. C.; COSTA, A. C. R. How Planning Becomes Improvisation? – A Constraint Based Approach for Director Agents in Improvisational Systems. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 16., 2002. **Proceedings...** Berlin: Springer, 2002.
- [MOR 2003a] MORAES, M. C.; BERTOLETTI, A. C.; BERTOLETTI, A. C.; TRINTIN, J. E. G. Conceção, Desenvolvimento e Novos Estudos do Museu Virtual SAGRES. In: WORKSHOP DE MUSEUS E CENTROS DE CIÊNCIAS: DESIGN E CONSTRUÇÃO DE EXPERIMENTOS, 2., 2003. **Anais...** Porto Alegre:[s.n.], 2003.
- [MOR 2003b] MORAES, M. C.; COSTA, A. C. R. Structuring the Coordination Space of Multiagent Systems. In: CONFERENCIA INTERNACIONAL DE LA SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACIÓN, 23., 2003. **Proceedings...** Chillán/Chile: [s.n.], 2003.
- [MOR 2003c] MORAES, M. C.; COSTA, A. C. R. Estruturando o Espaço de Coordenação de Sistemas Multiagentes. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 4., 2003. **Anais...** Campinas, São Paulo: UNICAMP:SBC, 2003.
- [MOR 2003d] MORAES, M. C.; COSTA, A. C. R. Proposta de um Espaço de Coordenação Tridimensional para Sistemas Multiagentes. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 4., 2003. **Anais...** Campinas, São Paulo: UNICAMP:SBC, 2003.
- [MOR 74] MORRISON, H. **Directing in The Theatre**. New York: Theatre Arts Books, 1974.
- [MOU 96] MOULIN, B.; CHAIB-DRAA, B. An Overview of Distributed Artificial Intelligence. In: O'HARE, G.; JENNINGS, N. (Ed.). **Foundations of Distributed Artificial Intelligence**. New York: John Wiley & Sons, 1996. p. 3- 56.

- [MUL 98] MULKEN, S. van; ANDRÉ, E.; MÜLLER, J. The Persona Effect: How Substantial Is It? In: HUMAN COMPUTER INTERACTION, 13., 1998. **Proceedings...** Berlin: Springer, 1998. p.53-66. Disponível em: <<http://www.dfki.de/~mulken/pubs.html>>. Acesso em: jan. 2000.
- [MÜL 96] MÜLLER, H. J. Negotiation Principles. In: O'HARE, G.; JENNINGS, N. (Ed.). **Foundations of Distributed Artificial Intelligence**. New York: John Wiley & Sons, 1996. p. 211 – 229.
- [NIC 2002] NICKLES, M.; ROVATSOS, M.; WEISS, G. A Schema for Specifying Computational Autonomy. In: INTERNATIONAL WORKSHOP ENGINEERING SOCIETIES IN THE AGENTS WORLD, 3., 2002. **Proceedings...** Disponível em: <www.ai.univie.ac.at/~paolo/conf/esaw02/index.html>. Acesso em: jan. 2002.
- [NWA 96] Nwana, H.S.; LEE, L.; JENNINGS, N. R. Co-ordination in software agent systems. **The British Telecom Journal**, [S.l.], v.14, n.4, p. 79-88, 1996.
- [OLI 2001] OLIVERIA, E.; ROCHA, A. P. **Agents advanced features for negotiation in electronic commerce and virtual organizations formation process**. Berlin:Springer, 2001. p. 77-96. (Lecture Notes in Computer Science, 1991).
- [ORT 2000] ORTIZ, C.; HSU, E.; DESJARDIN, M. Incremental Negotiation and Coalition Formation for Resource-bounded Agents Preliminary Report. **Autonomous Agents and Multi-Agent System**, Amsterdam, v. 5, n. 4, p. 429-465, 2000.
- [PAN 2001] PANZARASA, P.; JENNINGS, N. Negotiation and joint commitments in multi-agent systems. In: INT. WORKSHOP ON MODELLING ARTIFICIAL SOCIETIES AND HYBRID ORGANISATIONS, 2., 2001. **Proceedings...** Vienna, Austria:[s.n.], 2001.
- [PAP 98] PAPADOPOULOS, G. A.; ARBAB, F. Coordination Models and Languages. **Advances in Computers 46: The Engineering of Large Systems**, Cambridge, v. 46, n. 1, p. 329-400, Aug. 1998.
- [PAR 98a] PARSONS, S.; JENNINGS, N. R. Argumentation and multi-agent decision making. In: AAAI SPRING SYMP. ON INTERACTIVE AND MIXED-INITIATIVE DECISION MAKING, 1998. **Proceedings...** Stanford: AAAI Press, 1998. p. 89-91.
- [PAR 98b] PARSONS, S.; SIERRA, C.; JENNINGS, N. R. Agents that reason and negotiate by arguing. **Journal of Logic and Computation**, Oxford, v.1, n. 3, p. 261-292, 1998.
- [PER 96a] PERLIN, K.; GOLBERG, A. **Improvisational Animation**. 1996. Disponível em: <<http://www.mrl.nyu.edu/improv/whitepaper/index.html>>.

Acesso em: mar. 2000.

- [PER 96b] PERLIN, K.; GOLBERG, A. Improv: A System for Scripting Interactive Actors in Virtual Worlds. **Computer Graphics**, New York, v. 29, 1996.
- [PFL 96] PFLEGER, K.; HAYES-ROTH, B. Plans should abstractly describe intended behavior. In: MEYSTEEL, A.; ALBUS, J.; QUINTERO, R. (Ed.). **INTELLIGENT SYSTEMS: A SEMIOTIC PERSPECTIVE. Proceedings...** Maryland: [s.n.], 1996. v.1, p.29-34. Disponível em: <<http://www-cs-students.stanford.edu/~kpfleger/publications>>. Acesso em: mar. 2000.
- [PFL 98] PFLEGER, K.; HAYES-ROTH, B. **Using Abstract Plans to Guide Behavior**. Stanford: Stanford University, 1998. (Technical Report, KSL-98-02).
- [POL 90] POLLACK, M. Plans as Complex Mental Attitudes. In: COHEN, P. R.; MORGAN, J.; POLLACK, M. E. (Ed.). **Intentions in Communication**. Cambridge: MIT Press, 1990.
- [POL 92] POLLACK, M. The Uses of plans. **Artificial Intelligence**, [S.l.], v.57, p. 43-68, 1992.
- [POL 99] POLLACK, M.; HORTY, J. There's More to Life than Making Plans: Plan Management in Dynamic, Multi-agent Environment. **The AI Magazine**, Amsterdam, v. 20, n. 4, p. 71-84, 1999.
- [RAO 91a] RAO, A. S.; GEORGEFF, M. P. Deliberation and Intentions. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 7., 1991. **Proceedings...** San Mateo: Morgan Kaufmann, 1991.
- [RAO 91b] RAO, A. S.; GEORGEFF, M. P. **Modeling Rational Agents within a BDI-Architecture**. Melbourne: Australian Artificial Intelligence Group, 1991. (Technical Report, 14).
- [RAO 95] RAO, A. S.; GEORGEFF, M. P. **BDI Agents: From Theory to Practice**. Melbourne: Australian Artificial Intelligence Group, 1995. (Technical Report, 56).
- [REI 92] REILLY, W.S.; BATES, J. **Building Emotional Agents**. Pittsburg: Carnegie Mellon University, 1992. (Technical Report, CMU-CS-92-143).
- [REI 96] REILLY, W. S. **Believable Social and Emotional Agents**. 1996. Tese de Doutorado (CMU-CS-96-138) – Carnegie Mellon University, Pittsburg.
- [REV 87] REVERBEL, Olga. **Teatro: uma síntese em atos e cenas**. [S.l.]: LPM, 1987.
- [RIC 2002] RICCI, A.; OMICINI, A.; DENTI, E. Activity Theory as a Framework for MAS Coordination. In: **INTERNATIONAL WORKSHOP**

- ENGINEERING SOCIETIES IN THE AGENTS WORLD, ESAW, 3., 2002. **Proceedings...** Madrid, Spain:[s.n.], 2002.
- [RIC 93] RICH, E.; KNIGHT, K. **Inteligência Artificial**. 2 ed. São Paulo: Makron Books, 1993.
- [ROS 97] ROSENCEIN, J.; ZLOTKIN, G. Designing Conventions for Automated Negotiation. In: HUHNS, M.; SINGH, M. (Ed.). **Readings in agents**. San Francisco, CA : Morgan Kaufmann, 1997.
- [RUS 86] RUSSELL, S. A Quantitative Analysis of Analogy by Similarity. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 5., 1986. **Proceedings...** Philadelphia, PA: Morgan Kaufmann, 1986.
- [RUS 88] RUSSELL, S. Analogy by Similarity. In: HELMAN, D. (Ed.). **Analogical Reasoning**, Boston, MA: D. Reidel, 1988.
- [SAL 2001] SALVUCCI, D. D.; ANDERSON, J. R. Integrating Analogical Mapping and general problem solving: the path-mapping approach. **Cognitive Science**, Indiana, v. 25, p. 67-110, 2001.
- [SAN 95] SANDHOLM, T.; LESSER, V. Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, ICMAS, 1., 1995. **Proceedings...** [S.l.:s.n.], 1995. p. 328-335.
- [SIE 97] SIERRA, C.; FARATIN, P.; JENNINGS, N. A Service-Oriented Negotiation Model between Autonomous Agents. In: EUROPEAN WORKSHOP ON MODELING AUTONOMOUS AGENTS IN A MULTIAGENT WORLD, MAAMAW, 8., 1997. **Proceedings...** [S.l.:s.n.], 1997.
- [SIN 94] SINGH, M. **Multiagent Systems: A Theoretical Framework for Intentions, Know-How and Communications**. Berlin, Germany: Springer-Verlag, 1994. (Lecture Notes in Artificial Intelligence, 799).
- [SMI 89] SMITH, S.; BATES, J. **Towards a Theory of Narrative for Interactive Fiction**. Pittsburg: Carnegie Mellon University, 1989. (Technical Report, CMU-CS-89-121).
- [SOW 2003] SOWA, J. F.; MAJUMDAR, A. K. Analogical Reasoning. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL STRUCTURES. 2003. Disponível em: <<http://www.jfsowa.com/pubs/analog.htm>>. Acesso em: jun. 2003.
- [SPO 63] SPOLIN, V. **Improvisation for the Theater: A Handbook of Teaching and Directing Techniques**. [S.l.]: Northwestern University Press, 1963.
- [SPO 99] SPOLIN, V. **Improvisation for the Theater**. 3rd ed. [S.l.]: Northwestern University Press, 1999.

- [STE 95] STEFIK, M. **Introduction to Knowledge Systems**. [S.l.]: Morgan Kaufmann, 1995.
- [STO 96] STONE, B. A.; LESTER, J.C. **Dynamically Sequencing an Animated Pedagogical Agent**. 1996. Disponível em: <<http://www.csc.ncsu.edu/eos/users/l/lester/www/imedia/papers.html>>. Acesso em: abr. 2000.
- [SYC 96] SYCARA, K.; ZENG, D. Coordination of Multiple Intelligent Software Agents. **International Journal of Cooperative Information Systems**, New Jersey, v. 5, n. 2 e 3, p. 181-212, 1996.
- [SUC 87] SUCHMAN, L. A. **Plans and Situated Actions: The problem of human machine communication**. [S.l.:s.n.],1987.
- [TOL 2000] TOLKSDORF, R. Models of Coordination. In: OMICINI, A.; TOLKSDORF, R.; ZAMBONELLI, F. (Ed.). **Engineering Societies in The Agents World, 2000. Proceedings...** [S.l.]: Springer-Verlag, p. 78-92. (Lecture Notes in Artificial Intelligence, 1972).
- [WEG 97] WEGNER, P. Why Interaction Is More Powerful Than Algorithms. **Communications of the ACM**, New York, v. 40, n. 5, May 1997.
- [WEI 97] WEYHRAUCH, P. **Guiding Interactive Drama**. 1997. Tese de Doutorado (CMU-CS-97-109) – Carnegie Mellow University, Pittsburg.
- [WEI 99] WEISS, G. **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence**. Cambridge: The MIT Press. 1999.
- [WOO 98] WOOLDRIDGE, M. J.; JENNINGS, N. R. Pitfalls of Agent-Oriented Development. In: INT. CONF. ON AUTONOMOUS AGENTS, AGENTS, 2., 1998. **Proceedings...** Minneapolis, USA:AAAI Press, 1998. p. 385-391.
- [WOO 99] WOOLDRIDGE, M. Intelligent Agents. In: WEISS, G. (Ed.). **Multiagent systems: A Modern Approach to Distributed Artificial Intelligence**. Cambridge: The MIT Press, 1999. p. 27-77.
- [WOO 2000] WOOLDRIDGE, M. **Reasoning about Rational Agents**. Cambridge, Massachusetts: The MIT Press, 2000.
- [WUR 98] WURMAN, P.; WELLMAN, M.; WALSH, W. The Michigan Internet Auctionnot: A Configurable Auction Server for Human and Software Agents. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 1998. **Proceedings...** [S.l.]: ACM Press, 1998. p. 301 – 308.
- [VAR 99] VARELA, C.; AGHA, G. A Hierarquical Model for Coordination of Concurrent Activities. In: INTERNATIONAL CONFERENCE ON COORDINATION LANGUAGES AND MODELS, 3., 1999. **Proceedings...** Amsterdam, The Netherlands: Springer-Verlag, 1999. p. 166-182. (Lecture Notes in Computer Science, 1594).

- [VUL 2000] VULKAN, N.; JENNINGS, N. R. Efficient Mechanisms for the Supply of Services in Multi-Agent Environments. **Journal of Decision Support Systems**, New Jersey, v. 28, n. 1-2, p. 5-19, 2000.
- [ZHA 2000] ZHANG, X.; PODOROZHNY, R.; LESSER, V. Cooperative, MultiStep Negotiation Over a Multi-Dimensional Utility Function. In: IASTED INTERNATIONAL CONFERENCE, ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING, ASC, 2000. **Proceedings...** [S.l.:s.n.], 2000. p. 136-142.
- [ZHA 2001] ZHANG, X. Q.; LESSER, V.; PODOROZHNY, R. New Results on Cooperative, MultiStep Negotiation Over a Multi-Dimensional Utility Function. In: AAAI FALL SYMPOSIUM ON NEGOTIATION METHODS FOR AUTONOMOUS COOPERATIVE SYSTEMS, 2001. **Proceedings...** [S.l.:s.n.], 2001.
- [ZLO 89] ZLOTKIN G.; ROSENSHEIN, J. Negotiation and Task Sharing among Autonomous Agents in Cooperative Domains. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 11., 1989. **Proceedings...** [S.l.]: Morgan Kaufmann, 1989. p. 912-917.
- [ZLO 90] ZLOTKIN G., ROSENSHEIN, J. Negotiations and conflict resolutions in non-cooperative domains. In: AAAI, 1990. **Proceedings...** Boston, MA: AAAI Press, 1990. p. 100-105.
- [ZLO 96a] ZLOTKIN, G.; ROSENCEIN, J. Mechanisms for Automated Negotiation in State Oriented Domains. **Journal of Artificial Intelligence Research**, Washington, v.5, p. 163-238, 1996.
- [ZLO 96b] ZLOTKIN, G.; ROSENCEIN, J. Mechanism design for automated negotiation and its application to task oriented domains. **Artificial Intelligence**, [S.l.], v.86, p. 195- 244, 1996.