JAIME SIMÃO SICHMAN

Raciocínio Social e Organizacional em Sistemas Multiagentes: Avanços e Perspectivas

Tese apresentada à Escola Politécnica da Universidade de São Paulo como requisito para obtenção do Título de Professor Livre Docente, junto ao Departamento de Engenharia de Computação e Sistemas Digitais.

JAIME SIMÃO SICHMAN

Raciocínio Social e Organizacional em Sistemas Multiagentes: Avanços e Perspectivas

Tese apresentada à Escola Politécnica da Universidade de São Paulo como requisito para obtenção do Título de Professor Livre Docente, junto ao Departamento de Engenharia de Computação e Sistemas Digitais.

Especialidade: Inteligência Artificial

Ficha Catalográfica

Sichman, Jaime Simão

Raciocínio Social e Organizacional em Sistemas Multiagentes: Avanços e Perspectivas. São Paulo, 2003. 235 p.

Tese (Livre Docência) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Computação. 2. Inteligência Artificial. 3. Sistemas Multiagentes. 4. Raciocínio Social em Sistemas Multiagentes. 5. Raciocínio Organizacional em Sistemas Multiagentes. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. Título.

Longe se vai
Sonhando demais
Mas onde se chega assim
Vou descobrir
O que me faz sentir
Eu caçador de mim

(Sérgio Magrão / Luiz Carlos Sá)

AGRADECIMENTOS

Aos meus colegas, docentes e funcionários, da Universidade de São Paulo (USP), da centenária Escola Politécnica (EP), e mais particularmente do Departamento de Engenharia de Computação e Sistemas Digitais (PCS), pelo ambiente estimulante de trabalho que me proporcionaram desde o início de minha carreira. Gostaria que todos estes se sentissem representados na homenagem que presto à pessoa do Prof. Antonio Marcos de Aguirra Massola, ex-chefe de nosso Departamento e ex-diretor da Escola Politécnica, que num longínquo dia de 1996 me convidou, e porque não dizer me seduziu, a ingressar na carreira acadêmica, demonstrando a sua confiança na minha capacidade, à qual espero sinceramente ter correspondido ao longo de todos estes anos.

Aos professores, colaboradores e alunos, integrantes e ex-integrantes do Laboratório de Técnicas Inteligentes (LTI), por saberam construir, cotidianamente nos últimos cinco anos, um ambiente científico crítico e rigoroso, mas também agradável e acolhedor, pois fazer ciência deve ser, antes de tudo, uma atividade prazeirosa, e porque não dizer lúdica.

Aos meus colegas pertencentes à comunidade brasileira e internacional de pesquisa em Inteligência Artificial, e mais particularmente em Sistemas Multiagentes, pelas inúmeras oportunidades de momentos de reflexão conjuntos que me proporcionaram nos últimos anos, quando nos encontramos nos diversos Congressos, reuniões e comissões que fazem parte do dia-a-dia de nosso ofício. Gostaria de ressaltar particularmente os grupos de pesquisa vinculados ao Departamento de Informática da Universidade de Lisboa (Portugal), à Ecole Nationale Supérieure de Mines de Saint-Etienne (França) e ao Istituto di Scienze e Tecnologie della Cognizione (Itália), com os quais mantive uma profícua cooperação durante os últimos anos, cujos resultados se encontram descritos neste trabalho. Cabe aqui ainda um agradecimento especial a dois colegas que pertencem a estes grupos, Dr. Olivier Boissier e Dra. Rosaria Conte, com os quais tive a chance de partilhar não somente as suas inegáveis qualidades como cientistas e pesquisadores,

como também seu calor humano e amizade. Que saibam que há muito tempo se encontram no rol de meus amigos mais preciosos.

Aos Profs. Helder Coelho e Cristiano Castelfranchi, que por suas qualidades humanas, éticas, morais e científicas, continuam a desempenhar o papel de farol e exemplo no qual humildemente me espelho.

A Jomi Fred Hübner, pelo imenso trabalho que teve em desenvolver o estilo LETEX utilizado para a geração deste manuscrito.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), por terem possibilitado o desenvolvimento da maioria dos trabalhos aqui citados, através da concessão de financiamentos.

Aos meus amigos, que pela terceira vez (e espero que seja a última!) se privaram de meu convívio durante o tempo de redação deste manuscrito.

A toda minha família, da qual sempre recebi o maior apoio e carinho, inclusive a meus queridos tios e avós que já se foram há tanto tempo, mas que continuam sempre presentes por meio dos que aqui ficaram.

A Rosa e Abrão Sichman, meus queridos pais, guias desta longa estrada e alicerces do que sou, a quem exprimo mais uma vez toda a minha gratidão e todo o meu amor.

A Isabella Giardulli Sichman, doce e guerreira companheira desta longa jornada, meu agradecimento, meu amor e minha promessa (?) de que esta é realmente a última tese ...

Finalmente, gostaria de ressaltar o meu mais profundo agradecimento e reconhecimento aos meus alunos e ex-alunos, de Iniciação Científica, Mestrado e Doutorado, sem a contribuição dos quais este trabalho certamente não teria sido viabilizado. Como esta tese trata da interação e organização de agentes autônomos, sua concretização, especialmente deste manuscrito, foi realmente uma atividade cooperativa, realizada por uma coalizão. Jomi, Gustavo, Nuno, Maria das Graças e Diana, membros desta coalizão, o objetivo global foi atingido. Esta tese também é de vocês!

RESUMO

Este trabalho apresenta um conjunto de teorias e aplicações que permitem validar a idéia de que agentes cognitivos autônomos, imersos num sistema multiagentes aberto, podem aumentar sua eficiência e adaptabilidade a alterações no ambiente, caso tenham a capacidade de representar explicitamente e explorar, através de mecanismos de raciocínio adequados, as capacidades de outros agentes e as eventuais organizações em que estejam envolvidos.

Foi demonstrado também que as interações entre estes agentes e as suas organizações formam um círculo virtuoso. Por uma lado, as interações podem eventualmente criar organizações dinâmicas, denominadas coalizões, propiciando uma ação coletiva. Por outro lado, caso tal ação coletiva deva se repetir com freqüência, torna-se mais adequado criar organizações formais, que limitem as interações entre os agentes, garantindo deste modo que estes atinjam seus objetivos globais de modo otimizado.

ABSTRACT

This work presents a set of theories and applications that validate the idea that autonomous cognitive agents, immersed in an open multi-agent system, can increase their efficiency and adaptability to changes in the environment, if they are provided with the ability to explicitly represent and exploit, through adequate reasoning mechanisms, the capacities of other agents and the eventual organizations of which they make part.

It is also demonstrated that the interactions among these agents and their organizations do form a virtuous circle. On one hand, the interactions can eventually create dynamic organizations, called coalitions, to foster a collective action. On the other hand, if the collective action is to be repeated frequently, it becomes more adequate to create formal organizations that will limit the agents' interactions to ensure that they will reach their global goals in an optimized way.

Sumário

1	INT	RODUÇÂ	ĂO	1
	1.1	Motiva	ação	1
	1.2	Racio	cínio social e organizacional	2
	1.3	Objeti	ivos	4
	1.4	Panor	ama de atuação	4
	1.5	Organ	ização do manuscrito	7
Ι	Fu	INDAM	ientos Básicos	Ö
2	Agi	ENTES E	E SISTEMAS MULTIAGENTES	10
	2.1	Agente	es	10
		2.1.1	Definição	11
		2.1.2	Taxonomia de agentes	12
		2.1.3	Agentes reativos e cognitivos	13
		2.1.4	Agentes autônomos	14
		2.1.5	Agentes racionais	15
		2.1.6	Arquiteturas de agentes	17
		2.1.7	Agentes e outros modelos computacionais	19
	2.2	Sistem	nas multiagentes	21
		2.2.1	Definição	21
		2.2.2	Taxonomia de sistemas multiagentes	22

		2.2.3	Histórico	23
		2.2.4	Domínios de aplicação	24
		2.2.5	Sistemas multiagentes e outros modelos computacionais	25
	2.3	Classi	ficação de pesquisas em sistemas multiagentes	28
		2.3.1	Resolução social e simulação social	28
		2.3.2	Teorias, arquiteturas e linguagens de agentes	29
		2.3.3	Agentes, ambientes, interações e organizações	30
	2.4	Conclu	usões	33
3	Inti	ERAÇÕE	es entre Agentes	34
	3.1	Interfe	erência social	34
	3.2	Situaç	ões de interação	35
		3.2.1	Interferência social neutra	36
		3.2.2	Interferência social positiva	36
		3.2.3	Interferência social negativa	37
		3.2.4	Dimensões complementares	38
	3.3	Repres	sentação das interações entre agentes	39
		3.3.1	Interação intencional	39
		3.3.2	Interação não intencional	40
	3.4	Model	o de resolução cooperativa de problemas	40
		3.4.1	Detecção do potencial de cooperação	41
		3.4.2	Formação de coalizão	41
		3.4.3	Formação de plano para a coalizão	41
		3.4.4	Ação da coalizão	42
	3.5	Métod	los de interação	42
		3.5.1	Agrupamento	42

		3.5.2	Especialização	42
		3.5.3	Divisão de tarefas e de resultados	43
		3.5.4	Coordenação de ações	44
		3.5.5	Resolução de conflitos	47
	3.6	Comu	nicação entre agentes	49
		3.6.1	Teoria dos atos de fala	50
		3.6.2	Linguagens de comunicação	51
		3.6.3	Ontologias de comunicação	53
		3.6.4	Protocolos de comunicação	54
	3.7	Conclu	usões	56
4	One		ções de Agentes	58
4				
	4.1	Defini	ção	58
	4.2	Caract	terísticas de organizações de agentes	59
	4.3	Finalio	dade de organizações de agentes	60
	4.4	Taxon	omia de organizações de agentes	61
	4.5	Gênes	e das organizações de agentes	61
		4.5.1	Organizações formais	62
		4.5.2	Organizações emergentes	62
	4.6	Foco d	das organizações de agentes	63
		4.6.1	Aspectos estruturais	64
		4.6.2	Aspectos funcionais	65
		4.6.3	Aspectos deônticos	67
	4.7	Níveis	de organizações de agentes	68
		4.7.1	Organizações abstratas	68
		4.7.2	Organizações concretas	68

	4.8	Repres	sentação das organizações de agentes	69
	4.9	Reorga	anização	71
	4.10	Conclu	ısões	73
II	Μ	ODEL	os	74
5	Мог	DELOS I	de Raciocínio Social	75
	5.1	Model	o ${\cal S}$ RM	75
		5.1.1	Ação social	76
		5.1.2	Delegação e adoção de objetivos	77
		5.1.3	Mecanismo de raciocínio social	77
		5.1.4	Descrição externa	78
		5.1.5	Relações de dependência	79
		5.1.6	Redes de dependência	80
		5.1.7	Relações de OU-dependência e de E-dependência	80
		5.1.8	Relações de dependência mútua e recíproca	82
		5.1.9	Situações de objetivo	83
		5.1.10	Situações de dependência	83
		5.1.11	Coalizões baseadas em dependência	85
		5.1.12	Coalizões baseadas em dependência extendidas	87
		5.1.13	Coalizões baseadas em dependência e rede contratual	88
	5.2	Model	o $\mathcal{S}_{\mathrm{RM}^+}$	90
		5.2.1	Relações de dependência circulares	91
		5.2.2	Grafos de dependência	91
		5.2.3	Grafos de dependência reduzidos	92
		5.2.4	AMONG-dependência	93

		5.2.5	GROUP-dependência	. 96
		5.2.6	COLLECTIVE-dependência	. 98
	5.3	Concl	usões	. 99
6	Mo	DELOS	de Raciocínio Organizacional	102
	6.1	Model	o \mathcal{M} oise	. 102
		6.1.1	Definições preliminares	. 103
		6.1.2	Nível individual	. 104
		6.1.3	Nível social	. 106
		6.1.4	Nível coletivo	. 107
		6.1.5	Especificação e entidade organizacional	. 108
		6.1.6	Dependência organizacional	. 109
		6.1.7	Coerência organizacional	. 111
	6.2	Model	o \mathcal{M} oise $^+$. 113
		6.2.1	Organização e autonomia dos agentes	. 114
		6.2.2	Especificação estrutural	. 115
		6.2.3	Especificação funcional	. 116
		6.2.4	Especificação deôntica	. 119
		6.2.5	Especificação e entidade organizacional	. 120
		6.2.6	Reorganização	. 120
	6.3	Concl	usões	. 123
II	I A	APLIC.	AÇÕES	126
7	Амі	BIENTE	s de Desenvolvimento	127
	7.1	Ambie	ente ${\cal M}$ ASENV	. 127

		7.1.1	Implementando agentes como objetos ativos	128
		7.1.2	Sistema \mathcal{D}_{PSK+P}	129
		7.1.3	Comunicação entre agentes no ambiente \mathcal{M} ASENV	131
	7.2	Ambie	ente ${\cal S}$ ACI	133
		7.2.1	Entrada e saída de agentes	134
		7.2.2	Envio e recebimento de mensagens	134
		7.2.3	Anúncio de habilidades	134
		7.2.4	Exemplo de interação no ambiente \mathcal{S}_{ACI}	135
		7.2.5	Ambiente KSACI	136
	7.3	Ambie	ente ${\cal S}$ IMCOG	137
	7.4	Conclu	ısões	138
8	$ m A_{PL}$	ICAÇÕE	es em Raciocínio Social	140
	8.1	Aplica	ição ${\cal D}$ EPNET	140
	8.2	Aplica	ação ${\cal D}$ EPINT	144
		8.2.1	Objetivos do sistema	144
		8.2.2	Princípios sobre o comportamento dos agentes	145
		8.2.3	Arquitetura dos agentes	145
		8.2.4	Funcionamento do sistema	147
		8.2.5	Protocolo de formação de coalizão	148
	8.3	Aplica	ιςãο $\mathcal D$ ΕΡΙΝΤ $^+$	149
	8.4	Conclu	ısões	151
9	$ m A_{PL}$	ICAÇÕE	es em Raciocínio Organizacional	152
	9.1	Aplica	ção ${\mathcal R}$ AISORG	152
		9.1.1	Cálculo de dependências organizacionais	154

		9.1.2	Detecção de incoerências organizacionais	55
		9.1.3	Heurísticas para restabelecer a coerência organizacional 1	57
	9.2	Aplica	ção ${\cal C}$ OLLORG	58
	9.3	Aplica	ção ${\mathcal J}$ ОЈТЕАМ	61
	9.4	Aplica	ção ${\cal C}$ OMMUNET	63
	9.5	Conclu	ısões	66
10	Out	RAS A	PLICAÇÕES MULTIAGENTES 10	67
	10.1	Aplica	ção ${\cal P}$ ARTNET $^+$	67
	10.2	Aplica	ção ${\cal O}$ PTIMA	71
	10.3	Aplica	ções do projeto ${\cal N}$ Alamas	76
		10.3.1	Abordagem cognitiva	77
		10.3.2	Abordagem reativa	78
	10.4	Conclu	ısões	79
11	Con	CLUSÕI	ES 1	81
\mathbf{A}	NEXO	OS	18	33
A	Des	SCRIÇÃ	o Formal do Modelo \mathcal{S} rm	83
	A.1	Lingua	gem interna	83
		A.1.1	Noções básicas	83
		A.1.2	Noções sobre autonomia	84
		A.1.3	Relações de dependência	85
		A.1.4	Situações de objetivo	85
		A.1.5	Situações de dependência	85
		A.1.6	Planos realizáveis	86

		A.1.7	Objetivos atingíveis	
	A.2	Lingua	agem externa	;
		A.2.1	Noções básicas	,
		A.2.2	Descrições externas compatíveis	,
		A.2.3	Hipótese de compatibilidade de descrição externa 187	,
		A.2.4	Inconsistência no nível de agência	,
В	DE	scriçã	o Formal do Modelo \mathcal{S} rm $^+$ 188	,
	B.1	Grafos	s de dependência	,
	B.2	Grafos	s de dependência reduzidos)
	B.3	AMO	NG-dependência)
		B.3.1	AMONG-dependência e OU-dependência 190)
		B.3.2	AMONG-dependência e E-dependência)
	B.4	GROU	JP-dependência	
	B.5	COLL	ECTIVE-dependência	
С	DE	SCRIÇÃ	o Formal do Modelo <i>M</i> oise 192	;
	C.1	Defini	ções preliminares)
		C.1.1	Plano)
		C.1.2	Protocolo de comunicação	;
	C.2	Nível i	individual	:
		C.2.1	Papel	
		C.2.2	Missão	:
	C.3	Nível s	social	:
		C.3.1	Relação organizacional	:
	C.4	Nível (coletivo	

	C.4	.1 Grupo		195	
С	.5 Es	rutura organizacional		195	
С	.6 En	idade organizacional		195	
	С.	1.1 Papéis instanciados		196	
	С.	5.2 Instância de grupo		196	
D I	Descr	ição Formal do Modelo \mathcal{M} oise $^+$		197	
D	.1 Es	pecificação Organizacional		197	
D	.2 Es	pecificação Estrutural		197	
	D.:	2.1 Nível individual: papéis		197	
	D.:	2.2 Nível social: ligações		198	
	D.:	2.3 Nível coletivo: grupos		199	
D	.3 Es	pecificação funcional		201	
	D.	3.1 Metas globais		201	
	D.	3.2 Nível individual: missões		202	
	D.	3.3 Nível coletivo: esquemas		202	
	D.	3.4 Preferência entre missões		202	
D	.4 Es	pecificação deôntica		203	
D	.5 En	idade organizacional		204	
Reff	RÊNC	AS BIBLIOGRÁFICAS		205	
Apêndices 217					
I E	XEMP.	O DE CÓDIGO DE AGENTE NO AMBIENTE ${\cal M}$ ASENV		217	
TT T	Exemi	LO DE CÓDIGO DE AGENTE NO AMBIENTE SACI		220	

III	EXEMPLO	DE	SESSÃO	DE	UTILIZAÇÃO	DO	${\tt SISTEMA}$	\mathcal{D} EPINT	223
IV	Exemplo	DE	SESSÃO	DE	UTILIZAÇÃO	DO	SISTEMA	\mathcal{D} epin $ ext{T}^+$	230

LISTA DE FIGURAS

1.1	Raciocínio social e organizacional em sistemas multiagentes	5
2.1	Arquitetura de um agente BDI (WOOLDRIDGE, 2002)	18
2.2	Arquiteturas híbridas de agentes (WOOLDRIDGE, 2002)	19
2.3	Metodologia Vogais (CARDOZO; SICHMAN; DEMAZEAU, 1993)	31
2.4	Visão integrada da classificação de pesquisas em sistemas multiagentes	32
4.1	Ontologia organizacional (FOX et al., 1998).	63
4.2	Modelo organizacional Aaladin (FERBER; GUTKNECHT, 1998)	65
4.3	Modelo organizacional TAEMS (DECKER, 1996)	66
4.4	Representação intencional e não intencional de organizações formais e emergentes (HÜBNER, 2003)	70
5.1	Exemplo de rede de dependência no modelo \mathcal{S}_{RM} (SICHMAN; CONTE, 2002)	81
5.2	Comparação do fluxo global de mensagens entre as técnicas CBD e RC, para um sistema com 10 agentes (ITO; SICHMAN, 2000)	89
5.3	Exemplo de grafo de dependência no modelo \mathcal{S}_{RM}^+ (SICHMAN; CONTE, 2002)	92
5.4	Exemplo de grafo de dependência reduzido no modelo \mathcal{S}_{RM}^+ (SICH-MAN; CONTE, 2002)	93
5.5	Exemplo de AMONG-dependência (SICHMAN; CONTE, 2002)	94
5.6	Desigualdade em AMONG-dependência (SICHMAN; CONTE, 2002).	95
5.7	Incompatibilidade em AMONG-dependência (SICHMAN; CONTE, 2002)	95

5.8	AMONG-dependência frágil (SICHMAN; CONTE, 2002) 96
5.9	AMONG-dependência forte (SICHMAN; CONTE, 2002) 97
5.10	GROUP-dependência centralizada (SICHMAN; CONTE, 2002) 98
5.11	COLLECTIVE-dependência (SICHMAN; CONTE, 2002) 99
5.12	Classificação dos modelos de raciocínio social
6.1	Exemplo de estrutura e entidade organizacional no modelo \mathcal{M} OISE (HANNOUN, 2002)
6.2	Efeito da organização no comportamento dos agentes (HÜBNER, 2003)
6.3	Foco estrutural de uma especificação e de uma entidade organizacional no modelo \mathcal{M} OISE ⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c).117
6.4	Foco funcional de uma especificação e de uma entidade organizacional no modelo \mathcal{M} OISE ⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c).118
6.5	Especificação estrutural do processo de reorganização no modelo \mathcal{M} OISE ⁺ (HÜBNER, 2003)
6.6	Especificação funcional do processo de reorganização no modelo \mathcal{M} OISE ⁺ (HÜBNER, 2003)
6.7	Classificação dos modelos de raciocínio organizacional 124
7.1	Implementação do modelo de agente utilizando objetos ativos (CARDOZO; SICHMAN; DEMAZEAU, 1993)
7.2	Monitoração do conteúdo da memória compartilhada no sistema \mathcal{D}_{PSK+P} (CARDOZO; SICHMAN; DEMAZEAU, 1993) 132
7.3	Exemplo de anúncio de habilidades no ambiente SACI (HÜBNER; SICHMAN, 2000a)
7.4	Serviço de páginas brancas no ambiente SACI (HÜBNER; SICH-MAN, 2000a)
7.5	Exemplo de interação entre agentes no ambiente KSACI (ALBU-QUERQUE et al., 2002)

7.6	Classificação dos ambientes de desenvolvimento	139
8.1	Exemplo de interface do simulador \mathcal{D} EPNET (CONTE; SICHMAN, 1995)	141
8.2	Arquitetura de agente ASIC (SICHMAN, 1995)	146
8.3	Protocolo de comunicação do sistema \mathcal{D} EPINT (SICHMAN, 1995).	149
8.4	Protocolo de comunicação do sistema $\mathcal{D}\textsc{epint}^+$ (DAVID, 1998).	150
8.5	Interface do sistema $\mathcal{D}_{\text{EPINT}^+}$ (DAVID, 1998)	150
8.6	Classificação das aplicações de raciocínio social	151
9.1	Arquitetura do sistema \mathcal{C} OLLORG (HANNOUN, 2002)	159
9.2	Agente de interface do sistema \mathcal{C} OLLORG (HANNOUN, 2002)	160
9.3	Tela do simulador \mathcal{T} eambots (Teambots, 2000)	162
9.4	Interface de um agente mediador no sistema \mathcal{C} OMMUNET	165
9.5	Classificação das aplicações de raciocínio organizacional	166
10.1	Interface do simulador \mathcal{P} ARTNET $^+$	170
10.2	Exemplo de time assíncrono no sistema $\mathcal{O}_{\text{PTIMA}}$ (RABAK, 1999).	173
10.3	Interface com o usuário do sistema $\mathcal{O}{\rm PTIMA}$ (RABAK, 1999)	174
10.4	Visualizador do sistema \mathcal{O}_{PTIMA} (RABAK, 1999)	176
10.5	Abordagem cognitiva para PLN no projeto \mathcal{N} ALAMAS (SILVA; ABRAHÃO; LIMA, 1998)	
10.6	Abordagem reativa para PLN no projeto \mathcal{N} ALAMAS (PAIVA, 1997).	178
10.7	Classificação das outras aplicações multiagentes	179

LISTA DE TABELAS

2.1	Taxonomia de agentes (GARCIA; SICHMAN, 2003)	12
2.2	Taxonomia de sistemas multiagentes (GARCIA; SICHMAN, 2003)	23
3.1	Situações de interação (FERBER, 1995)	36
4.1	Taxonomia de organizações de agentes	61
5.1	Exemplo de descrição externa no modelo \mathcal{S}_{RM}	79
6.1	Exemplos de papéis e missões no modelo \mathcal{M} oise	106
6.2	Exemplo de especificação deôntica no modelo \mathcal{M} OISE ⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c)	120
6.3	Especificação deôntica do processo de reorganização no modelo \mathcal{M} OISE $^+$ (HÜBNER, 2003)	123
9.1	Exemplo de especificação organizacional para aplicação \mathcal{R} AISORG (HANNOUN, 2002)	153
9.2	Exemplo de entidade organizacional para aplicação \mathcal{R} AISORG (HANNOUN, 2002)	154
9.3	Heurísticas para restaurar a coerência organizacional (HANNOUN,	
	2002)	158
III.1		224

LISTA DE ABREVIATURAS

Δ	F A	\11tá	ìm.	ato	Fir	iito

CBD Coalizões Baseadas em Dependência

IA Inteligência Artificial

IAD Inteligência Artificial Distribuída

PLN Processamento de Linguagens Maturais

QAP Quadratic Assignment Problem

RC Rede Contratual

RPC Remote Procedure Call

SMA Sistema(s) Multiagentes

SOD Sistema(s) de Objetos Distribuídos

TSP Traveling Salesman Problem

1 Introdução

Este capítulo tem como objetivo situar o leitor no contexto deste trabalho. Inicialmente, apresenta-se na seção 1.1 um exemplo motivador para o estudo do raciocínio social e organizacional em sistemas multiagentes. A seguir, a importância destes dois aspectos em sistemas multiagentes abertos é ressaltada na seção 1.2. Os objetivos deste trabalho são então apresentados na seção 1.3. A seção 1.4 relata de forma resumida um panorama das atividades realizadas pelo autor neste domínio nos últimos 10 (dez) anos. Finalmente, a organização do manuscrito é apresentada na seção 1.5.

1.1 Motivação

Seja o seguinte cenário, adaptado de (ERCEAU; FERBER, 1991, pag. 751):

Oficina 7, tarde da noite. Clotaire, um robô móvel, deve transportar Millie de volta ao seu local habitual de trabalho. Millie se encontra na oficina 7 pois acabou de ser reparada por Justin e Jespard, dois outros robôs que Millie conhece bem. Millie pesa 200 kg, o que é um peso excessivo para Clotaire transportar, já que foi projetada para suportar uma carga máxima de até 120 kg. Ah, esta Millie! Sempre com problemas, sempre a lhe pedir favores! Ela, Clotaire, não tem competência específica alguma para transporte de máquinas. Trata-se de uma especialista em microsoldagem, e em transporte e manipulação de objetos finos e delicados. Entretanto, depois que transportou por inúmeras vezes vários lotes de peças para Millie, num dia em que os robôs transportadores estavam extremamente ocupados e que recusaram solenemente as solicitações de Millie, esta última vive lhe pedindo

ajuda. Felizmente, Berthold, 150 kg de carga útil e o menos especialista dos transportadores, não está sobrecarregado de trabalho e poderá lhe dar uma mão. Juntos, poderão transportar Millie, e aproveitar o trajeto para convencê-la de que ela deve pedir auxílio aos especialistas em transporte, até mesmo insistindo se necessário. Além do mais, dada a hora e o plano de trabalho do turno desta noite, Clotaire deverá ainda soldar algumas peças que William, um dos transportadores, lhe trouxe há uma hora atrás, no posto de trabalho 21.

Não se trata de ficção científica, mas simplesmente de um cenário¹ onde ocorrem interações entre um grupo de robôs autônomos, que possuem uma certa dose de "inteligência", e que cooperam para realizar seu trabalho.

O domínio da inteligência artificial distribuída (IAD), e mais particularmente dos sistemas multiagentes (SMA), se ocupa do estudo de modelos, arquiteturas computacionais e ambientes de programação capazes de realizar o cenário acima. Diferentemente dos estudos clássicos da Inteligência Artificial (IA), que se apóiam num modelo de inteligência centralizado, tais domínios se propõem a distribuir a inteligência entre vários agentes, não submetidos a um controle centralizado. Procura-se conceber sistemas multiagentes, onde cada elemento é dotado de certa autonomia e deve ser capaz de perceber, raciocinar, decidir, eventualmente aprender e agir num ambiente comum. Devido à presença de outros agentes, todo agente deve ser provido de certas capacidades de interação tais como coordenação de ações e negociação, no caso da ocorrência de eventuais conflitos.

1.2 RACIOCÍNIO SOCIAL E ORGANIZACIONAL

Ao se analisar o cenário apresentado na seção anterior, pode-se notar que os agentes representam conhecimento, raciocinam e atuam em 3 (três) níveis de abstração distintos, descritos a seguir e exemplificados através da atuação do robô Clotaire:

1. nível do domínio: Clotaire pode traçar um plano para atingir seu objetivo de soldar as peças que o robô William lhe trouxe;

¹Este cenário será utilizado como exemplo ao longo de todos os capítulos da parte I deste manuscrito, e referenciado como cenário de interação de robôs.

- 2. nível social: Clotaire conhece os robôs Justin e Jespard e sabe que ambos são capazes de reparar outros robôs;
- 3. *nível organizacional:* Clotaire sabe que não cabe a si a tarefa de carregar outros robôs, já que existem outros robôs cuja função na oficina é exatamente esta.

Observa-se então que neste tipo de cenário, um agente deve ter a capacidade de representar os outros agentes e a sua respectiva organização, além de explorar e revisar tais representações. Por explorar, entende-se utilizar tais representações em seus mecanismos de raciocínio e decisão, como por exemplo se negando a realizar alguma ação para a qual não esteja habilitado na organização ou enviando um pedido de auxílio a um outro agente somente se este último seja efetivamente capaz de realizar a ação desejada.

Cabe ressaltar neste ponto que as propriedades do cenário apresentado não se restringem à área de robótica móvel autônoma. Desde o início da década de 90, alguns autores como Tokoro (1993) e Hewitt (1993) já intuiam que os futuros sistemas de processamento de informação seriam compostos por vastas redes de processadores heterogêneos, nos quais grandes aplicações e bases de dados seriam executadas de forma decentralizada e autônoma. Tais autores chamaram na época estes ambientes de sociedades de objetos ou ainda organizações eletrônicas. Sem a menor sombra de dúvida, aplicações atuais que utilizam a Internet são exemplos mais do que convincentes de que tais sistemas já fazem parte da nossa realidade.

Estes sistemas são chamados de sistemas multiagentes abertos, pois a entrada e a saída de agentes do sistema ocorre de forma autônoma e dinâmica, sem um controle centralizado pré-definido. No cenário de interação de robôs, uma situação como esta corresponderia a um novo robô que chegasse para trabalhar ou caso o único robô capaz de realizar alguma tarefa, por exemplo a de limpeza, tivesse que sofrer uma manutenção preventiva.

Do ponto de vista dos requisitos técnicos para se conceber tais SMA abertos, necessita-se assegurar a interconexão, a interoperabilidade, a adaptação e a cooperação entre seus agentes (SICHMAN, 1995). As duas primeiras propriedades já se encontram razoavelmente bem sedimentadas, através da utilização de técnicas de redes de computadores, sistemas distribuídos e da adoção de padrões no pro-

1.3 Objetivos 4

jeto de software de tais sistemas, como por exemplo CORBA (NICOL; WILKES; MANOLA, 1993).

Este trabalho tem como um de seus focos ilustrar as duas últimas propriedades citadas acima. Por adaptação, entende-se que o comportamento de tais agentes deve se alterar devido à ocorrência de mudanças no ambiente em que estão inseridos, principalmente quando a disponibilidade de serviços se altera dinamicamente. Por outro lado, como tais serviços têm custos associados, deve-se prover um paradigma que possa garantir que tais agentes aceitem cooperar uns com os outros, dado que neste contexto a cooperação não pode ser tomada como uma hipótese pré-estabelecida.

1.3 Objetivos

Neste trabalho, defende-se a idéia de que um mecanismo de raciocínio social e um mecanismo de raciocínio organizacional são blocos constituintes básicos essenciais que possam tornar agentes autônomos aptos a agirem de modo adequado em SMA abertos, como mostra a figura 1.1. Um mecanismo é chamado de social quando utiliza informação sobre os outros para inferir novas informações a partir das informações correntes. Analogamente, chama-se de organizacional um mecanismo que utiliza informações sobre a ação coletiva dos agentes para os mesmos fins.

Como ficará mais claro na seqüência do manuscrito, a existência de tais mecanismos prevê que os agentes devam representar explicitamente, explorar durante suas atividades de raciocínio e decisão e eventualmente revisar certas propriedades dos outros agentes e da ação coletiva do grupo.

1.4 Panorama de atuação

O autor iniciou seus estudos na área de IAD durante seu programa de Mestrado, realizado na Escola Politécnica da USP entre 1987 e 1991. Sua dissertação, sob orientação do Prof. Eleri Cardozo, tratou do projeto e implementação de uma ferramenta de monitoração para um núcleo de resolução distribuída de problemas orientado a objetos (SICHMAN, 1991; SICHMAN; CARDOZO, 1992a). Tal núcleo,

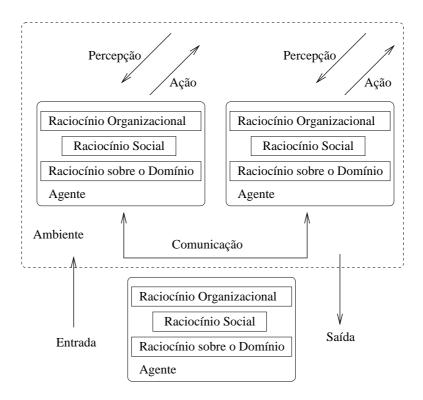


Figura 1.1: Raciocínio social e organizacional em sistemas multiagentes.

denominado \mathcal{D}_{PSK+P} (CARDOZO; SICHMAN, 1992), foi um dos pioneiros no país na utilização de objetos distribuídos, e serviu como base de desenvolvimento do ambiente \mathcal{M}_{ASENV} (CARDOZO; SICHMAN; DEMAZEAU, 1993).

No período de 1991 a 1995, iniciou sua especialização na área de SMA, durante seu programa de Doutorado junto ao Institut National Polytechnique de Grenoble (INPG), França. Trabalhou na equipe MAGMA, pertencente então ao Laboratoire d'Informatique Fondamentalle et Intelligence Artificielle (LIFIA). Esta equipe, liderada pelo seu orientador, Dr. Yves Demazeau, foi uma das pioneiras no estudo de SMA na Europa. O programa de Doutorado realizou-se ainda no âmbito europeu, tendo sido parte dele desenvolvido junto à equipe PSCS do Istituto di Psicologia del CNR, Roma, Itália. Tal equipe encontrava-se então sob a liderança do Dr. Cristiano Castelfranchi, atualmente um dos mais consagrados pesquisadores da área no mundo e com cuja equipe o autor mantém uma colaboração acadêmica até hoje. Sua tese propôs um modelo de raciocínio social baseado em redes de dependência (SICHMAN, 1995; SICHMAN et al., 1998) (SRM), inspirado na Teoria da Dependência proposta por Castelfranchi, Micelli e Cesta (1992). Como experimentação prática deste modelo, foram desenvolvi-

das duas aplicações, \mathcal{D} EPNET (CONTE; SICHMAN, 1995) e \mathcal{D} EPINT (SICHMAN, 1998).

Desde o seu retorno ao Brasil, ocorrido no final de 1995, o autor tem se dedicado a disseminar no país a área de pesquisa de SMA, através de palestras e minicursos. Juntamente com seus alunos de pós-graduação e colaboradores, realizou algumas extensões e avaliações de desempenho do modelo de raciocínio social (DAVID; SICHMAN; COELHO, 1999; ITO; SICHMAN, 2000) e propôs um novo modelo de raciocínio social, mais abrangente, baseado em grafos de dependência (S_{RM}^+) (CONTE; SICHMAN, 2002). Também participou do desenvolvimento de dois modelos de raciocínio organizacional, denominados Moise (HANNOUN et al., 2000) e Moise⁺ (HÜBNER; SICHMAN; BOISSIER, 2002b). Tais modelos sociais e organizacionais foram validados na prática através de alguns protótipos, finalizados ou em fase de desenvolvimento, denominados $\mathcal{D}_{\text{EPINT}^+}$ (DAVID; SI-CHMAN; COELHO, 2000), RAISORG (HANNOUN et al., 1998), COLLORG (HAN-NOUN, 2002), \mathcal{J} OJTEAM e \mathcal{C} OMMUNET. Também esteve envolvido na concepção de outras aplicações multiagentes, finalizadas ou em desenvolvimento, denominadas \mathcal{P} ARTNET⁺, \mathcal{O} PTIMA, e ainda duas aplicações resultantes do projeto temático Nalamas (Carvalho et al., 1998). Tais aplicações, embora não utilizem diretamente os modelos sociais e organizacionais acima citados, apresentam componentes que exploram o racicocínio social e organizacional dos agentes que as compõem. Para conceber alguns destes protótipos, foi necessário projetar e desenvolver alguns ambientes de desenvolvimento para SMA, denominados \mathcal{M} ASENV (CARDOZO; SICHMAN; DEMAZEAU, 1993), \mathcal{S} ACI (HÜBNER; SICHMAN, 2000a) (ambos finalizados) e $\mathcal{S}_{\text{IMCOG}}$ (MARIETTO et al., 2002) (em fase de desenvolvimento).

Como conseqüência destas atividades, já formou 1 (um) doutor (HANNOUN, 2002) e 4 (quatro) mestres (KARAM, 1995; DAVID, 1998; ITO, 1999; RABAK, 1999) na área. Além disto, participou em vários projetos de pesquisa nacionais e internacionais na área, alguns dos quais deram suporte financeiro aos projetos descritos no parágrafo anterior. No Brasil, foram realizadas cooperações com os grupos interessados em SMA do ICMC/USP, IME/USP, ITA/CTA, CIN/UFPE, II/UFRGS, DI/UFSC e II/PUC-RS. No exterior, foram realizadas cooperações com a CMU (Pittsburgh, EUA), ENSMSE (Saint-Etienne, França), CIRAD (Montpelier, França), FC/UL (Lisboa, Portugal) e IP/CNR (Roma,

Itália). Na quase totalidade deste projetos, sua atuação direta ou através da orientação de alunos foi dirigida às atividades de modelagem e desenvolvimento relativas ao raciocínio social e organizacional dos agentes presentes nos sistemas desenvolvidos.

1.5 Organização do manuscrito

Este manuscrito contém 11 (onze) capítulos, divididos em 3 (três) partes, além de 4 (quatro) anexos e 4 (quatro) apêndices.

A parte I apresenta um rápido painel da área de IAD e SMA, com ênfase nos tópicos de interações e organizações de agentes, fundamentais para a compreensão deste trabalho. No capítulo 2, apresenta-se um estado da arte da área de SMA, onde encontra-se definido o que se entende aqui por agentes e sistemas multiagentes, bem como duas taxonomias para caracterizar tais noções. Ao final deste capítulo, é apresentada uma proposta para classificar as pesquisas realizadas na área de SMA, utilizada em todos os outros capítulos para que o leitor possa melhor situar qual é a exata contribuição de cada um dos modelos e aplicações desenvolvidas pelo autor e seus colaboradores. O capítulo 3 apresenta uma outra taxonomia que permite classificar as principais situações de interação entre agentes, bem como os possíveis métodos de interação que podem ser utilizados em sua ação coletiva. Ao final deste capítulo, são discutidos também os protocolos de comunicação, cuja função é a de operacionalizar os métodos de interação anteriormente descritos. Já no capítulo 4, o conceito de organizações de agentes é explorado, através do uso de uma taxonomia onde se caracterizam sua gênese, seu foco de interesse, seu nível de descrição e seu comportamento. Alguns conceitos muito utilizados no domínio, como emergência, papéis, tarefas, compromissos e reorganização são devidamente caracterizados. A relevância dos aspectos abordados nestes capítulos para os trabalhos apresentados nas partes seguintes é realçada ao final de cada seção pertinente.

A parte II apresenta os modelos sociais e organizacionais desenvolvidos pelo autor, por seus alunos e por seus colaboradores. Em particular, os modelos de raciocínio social (\mathcal{S}_{RM} , \mathcal{S}_{RM}^+) são discutidos no capítulo 5 e os modelos de raciocínio organizacional (\mathcal{M}_{OISE} , \mathcal{M}_{OISE}^+) no capítulo 6.

A parte III apresenta uma série de experimentos e aplicações computacionais desenvolvidos pelo autor, por seus alunos e por seus colaboradores, que visam ilustrar a aplicabilidade prática dos modelos apresentados na parte II. No capítulo 7, encontram-se descritas as principais características de alguns ambientes de desenvolvimento para SMA (Masenv, Saci, Simcog), finalizados ou em fase de desenvolvimento. Tais ambientes foram eventualmente utilizados como substrato para o desenvolvimento das aplicações de raciocínio social (Depnet, Depint, Depint⁺) discutidas no capítulo 8 e de raciocínio organizacional (Raisorg, Collorg, Jojteam, Communet), apresentadas no capítulo 9. No capítulo capítulo 10, são abordadas ainda outras aplicações multiagentes (Partnet⁺, Optima e aplicações do projeto Nalamas). Embora tais aplicações não utilizem diretamente os modelos sociais e organizacionais apresentados na parte II, elas ilustram outros usos de conceitos relativos às interações e organizações de agentes apresentados na parte I.

O capítulo 11 apresenta as conclusões deste trabalho e algumas perspectivas futuras.

Para um leitor mais interessado nos aspectos teóricos dos modelos apresentados na parte II, uma descrição formal dos mesmos é apresentada nos anexos A, B, C e D. Adicionalmente, para um leitor que deseja vislumbrar os aspectos de implementação e de utilização dos ambientes \mathcal{M} ASENV e \mathcal{S} ACI descritos no capítulo 7, alguns trechos de código de agentes muito simples são apresentados nos apêndices I e II. Finalmente, os resultados obtidos em uma sessão de utilização dos sistemas \mathcal{D} EPINT e \mathcal{D} EPINT⁺, descritos no capítulo 8, são exibidos respectivamente nos apêndices III e IV.

Parte I

Fundamentos Básicos

2 Agentes e Sistemas Multiagentes

Este capítulo tem como objetivo fornecer ao leitor uma visão do domínio de agentes e SMA. Inicialmente, apresentam-se na seção 2.1 os principais conceitos relacionados com a noção de agente. Quando imersos num ambiente comum, estes formam os chamados sistemas multiagentes, cujas idéias fundamentais são discutidas na seção 2.2. Apresenta-se ainda na seção 2.3 uma proposta para classificar as pesquisas realizadas na área de SMA. Tal classificação é utilizada em todos os demais capítulos para que o leitor possa melhor situar qual é a exata contribuição na área de cada um dos modelos e aplicações desenvolvidas pelo autor, por seus alunos e por seus colaboradores, descritos respectivamente nas partes II e III. Finalmente, a seção seção 2.4 apresenta as conclusões deste capítulo.

2.1 AGENTES

Definitivamente, o termo agente está na agenda da computação neste início de século XXI. De modo similar ao que ocorreu com o termo "objeto" na década de 90, existe um certo exagero, e até mesmo modismo, em sua utilização. Por outro lado, associam-se significados muito distintos ao termo, conforme sua utilização em domínios distintos da computação, tais como redes de computadores (e seus agentes SNMP) e programação para a Internet (e seus agentes de busca de informações).

Neste trabalho, será adotada uma noção de agente proveniente da área de IA, e mais particularmente da área de SMA. Tais agentes são usualmente chamados de agentes "inteligentes".

2.1 Agentes 11

2.1.1 Definição

Não existe, atualmente, uma definição universalmente aceita de agente. Assim, será adotada aqui uma definição abrangente, proposta por Ferber (1995, pag. 15):

Um agente é uma entidade física ou virtual (a) capaz de agir num ambiente; (b) capaz de se comunicar com outros agentes; (c) que é movida por um conjunto de inclinações (sejam objetivos individuais a atingir ou uma função de satisfação a otimizar); (d) que possui recursos próprios; (e) que é capaz de perceber seu ambiente (de modo limitado) (f) que dispõe (eventualmente) de uma representação parcial deste ambiente; (g) que possui competência e oferece serviços; (h) que pode eventualmente se reproduzir; (i) e cujo comportamento tende a atingir seus objetivos utilizando as competências e os recursos que dispõe e levando em conta sua percepção, suas representações internas e as comunicações que recebe.

Nota-se que a definição acima é extremamente genérica, podendo compreender tanto um robô móvel (entidade física) como um robô virtual (entidade virtual)¹. Por outro lado, essa definição também ressalta o seu caráter integrador de grande parte dos assuntos estudados, de forma até então isolada, na IA clássica. Não é portanto gratuito o fato de que o livro didático de IA que mais causou impacto na comunidade nos últimos anos, escrito por Russel e Norvig (1995), utiliza a noção de agente como paradigma integrador das técnicas de IA.

Outra definição, proposta por Wooldridge e Jennings (1995a), considera que agentes inteligentes podem ser caracterizados pela presença das seguintes capacidades:

- reatividade: ao perceber seu ambiente, são capazes de responder às mudanças que ocorrem num tempo adequado para satisfazer seus objetivos;
- pró-atividade: exibem um comportamento direcionado a objetivos ao tomar a iniciativa de tentar satisfazê-los;

¹E até um ser humano!

2.1 Agentes 12

• habilidade social: podem interagir com outros agentes (e eventualmente com humanos) para tentar satisfazer seus objetivos.

2.1.2 Taxonomia de agentes

Garcia e Sichman (2003) propuseram uma taxonomia na qual um agente é visto como um tipo especial de sistema computacional e que pode ser classificado segundo alguns eixos² tais como:

- Eixo de Atuação: um agente pode atuar de forma isolada (Isolada)³ ou interagindo com outros agentes (Social);
- Eixo Cognitivo: um agente pode conter um modelo de representação interna do ambiente e dos outros agentes baseado em estados mentais e um modelo racional de decisão (Agente Cognitivo) ou apenas agir baseado num modelo de reações aos estímulos provocados pelo ambiente (Agente Reativo);
- Eixo Ambiental: um agente individual pode atuar de modo restrito numa única máquina (Agente de "Desktop") ou pode ter um campo de atuação mais amplo, numa rede, numa Intranet ou até mesmo na Internet (Agente de Internet).

Esta taxonomia encontra-se representada de forma resumida na tabela 2.1.

Eixo	Tipos de Agentes						
atuação		SOC	cial		individual		
cognitivo	reati	VO	cognitivo		cognitivo		
ambiental	"desktop"	Internet	"desktop"	Internet	"desktop"	Internet	

Tabela 2.1: Taxonomia de agentes (GARCIA; SICHMAN, 2003)

O foco deste trabalho são os agentes sociais. Quando reunidos no seio de um sistema, como mostrado na seção 2.2, permitem que este último apresente um comportamento "inteligente". Este comportamento, entretanto, pode surgir mesmo quando os agentes forem entidades muitos simples. É exatamente esta noção que é expressa no eixo cognitivo citado acima, e que é detalhado a seguir.

²Apenas os eixos relevantes para este trabalho são aqui apresentados.

³Neste caso, o agente interage apenas com os usuários humanos.

2.1 Agentes 13

2.1.3 Agentes reativos e cognitivos

Os agentes reativos são baseados em modelos de sociedades biológicas ou etológicas, como sociedades de abelhas e de formigas. Nestas sociedades, pode-se observar a emergência de um comportamento social dito "inteligente". As formigas, por exemplo, parecem ter uma certa estratégia coletiva inteligente quando estão em busca de alimento, mesmo que não se possa associar a propriedade de inteligência a uma formiga individualmente.

Os agentes reativos funcionam segundo um modelo de estímulo-resposta. Sua representação do ambiente e dos outros agentes não é explícita, não possuem memória do passado nem capacidade de planejar o futuro. Os agentes interagem indiretamente através da percepção de modificações no ambiente, que podem ter sido causadas como efeito das ações de outros agentes. Este é o caso, por exemplo, do traço químico deixado pelas formigas quando encontram uma fonte de alimento. Normalmente, as sociedades de agentes reativos são formadas por um número muito grande de agentes, da ordem de milhares. Agentes reativos estão relacionados com a capacidade de reatividade, discutida na seção 2.1.1.

Já os agentes cognitivos são baseados em modelos de sociedades humanas, como grupos, hierarquias e mercados. Possuem uma representação explícita do ambiente e dos outros agentes, usualmente possuem memória do passado e são capazes de planejar o futuro. Interagem diretamente, através do envio de mensagens. Podem eventualmente aprender com o resultado de suas ações. Tipicamente, sociedades de agentes cognitivos são compostas por uma ou duas dezenas de agentes. Agentes cognitivos estão relacionados com a capacidade de pró-atividade, discutida na seção 2.1.1.

Relevância: Nas partes II e III, são apresentados modelos e aplicações compostos por agentes cognitivos⁴, sociais e que atuam tanto em rede como numa máquina isolada.

 $^{^4}$ Com a exceção de um sistema desenvolvido no projeto \mathcal{N} ALAMAS, descrito na seção 10.3, onde se utiliza um modelo de agentes reativos.

2.1.4 AGENTES AUTÔNOMOS

Do mesmo modo como ocorre com o termo agente, existem diversos significados associados à noção de *autonomia* na literatura de IA e SMA. Podemos citar pelos menos 4 (quatro) significados distintos:

- 1. autonomia em relação à sua concepção: um agente é dito autônomo caso exista (ou tenha sido concebido) independentemente da existência de outros agentes (DEMAZEAU; MÜLLER, 1990). Tal visão é útil para se compreender uma das motivações do desenvolvimento de SMA segundo uma perspectiva de Engenharia de Software, ou seja, da concepção decentralizada e independente de subsistemas, que possam ser facilmente integrados;
- 2. autonomia em relação a seu ambiente: sinônimo de adaptativo, um agente é dito autônomo caso possa sobreviver num ambiente dinâmico e incerto, que seja percebido de forma incompleta, que possa se modificar independentemente do controle do agente e sobre o qual os efeitos das ações que o agente possa realizar não são previsíveis. Tal visão é tipicamente aquela adotada em trabalhos de robótica móvel (NILSSON, 1994);
- 3. autonomia em relação à execução de seus objetivos: sinônimo de autosuficiente, um agente é dito autônomo caso possa atingir seus objetivos sem a ajuda de outros agentes. Esta noção é utilizada na Teoria da Dependência (CASTELFRANCHI; MICELLI; CESTA, 1992; SICHMAN, 1995) para caracterizar a interação entre agentes;
- 4. autonomia em relação à suas motivações: sinônimo de não benevolente, um agente é dito autônomo caso tenha o livre-arbítrio de interagir socialmente. Através do conteúdo de suas crenças e de seu estado mental, pode decidir se deseja ou não cooperar com os outros.

Algumas destas noções são utilizadas nas seções 2.1.7 e 2.2.5 para diferenciar agentes e SMA de outros modelos computacionais.

Relevância: Neste trabalho, quando não sinalizado explicitamente, é adotado o último significado apresentado acima, ou seja, um agente tem controle completo de seu estado mental e decide as ações que deseja realizar.

2.1.5 AGENTES RACIONAIS

Uma questão fundamental no estudo de agentes cognitivos é o estabelecimento de modelos precisos de comportamento, que possam garantir que a boa decisão seja tomada no momento correto.

Supondo que um agente isolado tenha vários objetivos a atingir, qual dentre estes objetivos ele escolhe atingir em primeiro lugar? Como constrói um plano para isto? Caso tenha mais de um plano pré-estabelecido, qual deve ser utilizado? Quais critérios devem ser levados em conta para estas escolhas?

No caso de agentes sociais, além destas dimensões individuais, um componente adicional entra em cena, relativo à escolha de outros agentes para uma ação coletiva. Porque um agente deve aceitar colaborar com um outro? Por que deve desejar interagir com os outros agentes?

Existem basicamente dois modelos distintos de *racionalidade*, descritos a seguir, que tentam responder a estas questões.

2.1.5.1 Modelos baseados na utilidade

Estes modelos tomam como princípio da racionalidade o *utilitarismo*. Segundo esse paradigma, um agente é dito racional caso sempre busque a maximização de sua utilidade esperada (DOYLE, 1992). Essa noção de racionalidade encontra-se presente na maior parte das teorias econômicas modernas.

Como mostrado em (SICHMAN, 1995), os elementos de base do utilitarismo são a Teoria da Utilidade e a Teoria da Decisão. A Teoria da Utilidade propõe um modelo para caracterizar as preferências de um agente entre diversos estados possíveis do mundo, refletindo seus desejos de alto nível. A Teoria da Decisão estende este modelo, permitindo representar operações quantitativas sobre as preferências (por exemplo, a soma) e também tornando possível o cálculo das preferências na presença de incertezas (utilidade esperada). A Teoria dos Jogos ("Game Theory") (AXELROD, 1984) é construída sobre essas duas teorias, se interessando sobre uma fonte particular de incertezas: aquela que leva em conta a ação de outros agentes, considerados igualmente como racionais (GMYTRASI-EWICZ; DURFEE, 1993).

Assim, nestes modelos um agente irá propor a outro uma ação coletiva (e, de modo semelhante, o outro aceitará fazer parte desta ação) somente se esta ação for preferível às outras ações possíveis no momento. Esta preferência é capturada quantitativamente por meio de uma função matemática (função utilidade).

Como inconvenientes de tais modelos, podemos citar o fato de considerarem os agentes como homogêneos e auto-suficientes, a incapacidade de explicarem a ocorrência de cooperação (vide o chamado "dilema do prisioneiro" (LOMBORG, 1993)), a ausência da noção de objetivos e de adotarem uma visão da ação social como puramente estratégica (CONTE; SICHMAN, 1995). Por outro lado, a grande vantagem da utilização de modelos utilitários está no fato de apresentarem um arcabouço matemático muito rigoroso e bem formalizado, facilitando a obtenção de modelos e resultados teóricos bastante complexos. Grande parte de trabalhos atualmente realizados em SMA são fundados em modelos utilitários. Uma referência fundamental sobre o seu uso em SMA são sem dúvida alguma os trabalhos apresentados em (ROSENSCHEIN; ZLOTKIN, 1994).

2.1.5.2 Modelos baseados na complementaridade

Nestes outros modelos, as escolhas dos agentes para interagir com outros são fundamentadas em relações estruturais objetivas (que existem na realidade) nas quais os agentes encontram-se mergulhados. Uma destas relações fundamentais é a relação de dependência social (CASTELFRANCHI; MICELLI; CESTA, 1992), que se encontra melhor detalhada na seção 5.1.5. Resumidamente, agentes quase sempre necessitam uns dos outros para atingirem seus objetivos, e quando estas relações de dependência tornam-se subjetivas (pertencentes ao estado mental dos agentes), elas podem explicar por que agentes adotam os objetivos uns dos outros e por quê algumas interações sociais surgem no seio de uma sociedade.

Relevância: Nos trabalhos discutidos nas partes II e III, foram utilizados prioritariamente agentes cujo modelo de racionalidade é baseado na complementaridade.

2.1.6 ARQUITETURAS DE AGENTES

Existem diversas arquiteturas de agentes propostas na literatura, tanto voltadas para caracterizar agentes reativos como destinadas a representar agentes cognitivos. Uma breve discussão de suas principais características é apresentada a seguir.

2.1.6.1 Arquiteturas reativas

Basicamente, as arquiteturas reativas surgiram no final da década de 80, como uma reação de alguns pesquisadores à assim chamada IA simbólica. Quase todas estas reações rejeitam o fato de que a inteligência deve ser baseada em manipulações de representações simbólicas e defendem a idéia que um conjunto de comportamentos simples pode, em conjunto, resolver problemas reais complexos. Sem dúvida alguma, uma arquitetura de grande impacto foi a chamada "subsumption architecture", proposta por Brooks (1986). Nesta arquitetura, utilizada num contexto de robótica móvel, vários comportamentos elementares (como evitar colisões e seguir linhas) são simultaneamente ativados num robô, cuja ação é uma soma de todos os comportamentos. Tais comportamentos são organizados em camadas, sendo que algumas destas camadas podem ativar ou inibir o comportamento de outras.

2.1.6.2 Arquiteturas deliberativas

Basicamente, as arquiteturas deliberativas seguem a tradição dos primórdios da IA, sendo fortemente baseadas no paradigma simbólico. Os agentes mantêm uma representação interna que captura o estado do mundo, incluindo o estado dos outros agentes⁵. Tais arquiteturas são bastante utilizadas para o desenvolvimento de agentes cognitivos. Dentre as mais utilizadas, destaca-se a denominada arquitetura BDI ("Belief, Desire, Intention"), surgida no Stanford Research Institute, na década de 80. Tal arquitetura é baseada na Teoria de Raciocínio Prático do filósofo Michael Bratman (BRATMAN; ISRAEL; POLLACK, 1988). Por raciocínio prático, entende-se o processo de decidir, minuto a minuto, qual ação um ser humano deve realizar para atingir seus objetivos. Esta arquitetura, que foca

⁵Restrito ao caso dos agentes cognitivos sociais.

particularmente no papel das *intenções* no raciocínio prático humano, encontrase representada na figura 2.1 (WOOLDRIDGE, 1999). De forma resumida, nesta arquitetura os agentes têm *crenças* sobre o estado do mundo, que fazem com que *desejos* sejam gerados. Um mecanismo de filtragem seleciona os desejos que podem ser concretizados, de acordo com o estado atual das suas crenças. Tais desejos se tornam *intenções*, que irão guiar as ações que o agentes irão realizar. As intenções também irão limitar a geração de novos desejos, pois intenções, ao contrário de desejos, devem ser consistentes. Intenções também apresentam outras propriedades, como por exemplo persistência, e são associadas à noção de compromisso, discutida na seção 4.6.3.

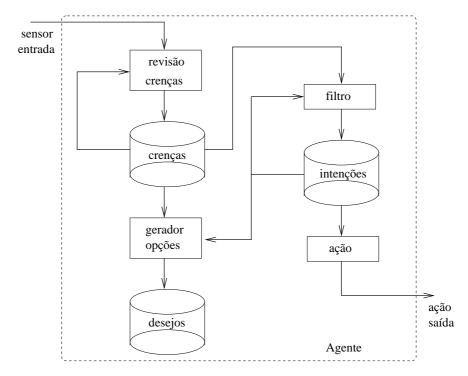


Figura 2.1: Arquitetura de um agente BDI (WOOLDRIDGE, 2002).

2.1.6.3 ARQUITETURAS HÍBRIDAS

Na prática, principalmente em sistemas de tempo real e em robótica, normalmente se deseja que um agente possa apresentar ambas as capacidades, reatividade e pró-atividade, que devem se manifestar conforme a ocasião assim o exija. Por exemplo, um robô móvel deve planejar sua trajetória, mas deve evitar rapidamente eventuais obstáculos que se lhe apresentem no caminho. Assim, criaramse arquiteturas híbridas, que podem ser representadas por camadas reativas ou

cognitivas interrelacionadas, como mostra a figura 2.2 (WOOLDRIDGE, 2002). Tais arquiteturas podem ser horizontais⁶ (ou baseadas em comportamentos) ou verticais (ou hierárquicas), combinando camadas reativas e cognitivas. Um exemplo destas arquiteturas é a chamada TouringMachine (FERGUSON, 1992), uma arquitetura horizontal composta por uma camada reativa, que provê respostas imediatas a eventuais mudanças no ambiente, uma camada de planejamento, responsável pelo comportamento pró-ativo, e uma camada que modela o ambiente e os outros agentes, e que gera novos objetivos para solucionar eventuais conflitos.

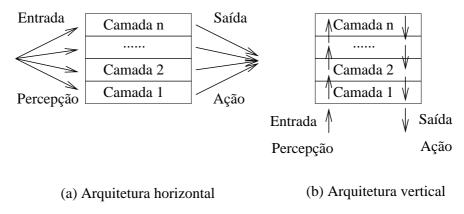


Figura 2.2: Arquiteturas híbridas de agentes (WOOLDRIDGE, 2002).

Relevância: No conjunto das aplicações realizadas no contexto deste trabalho, descritas na parte III, foram utilizadas basicamente arquiteturas deliberativas para os agentes que compõem as diversas aplicações.

2.1.7 Agentes e outros modelos computacionais

Para finalizar esta seção, discutem-se abaixo as principais diferenças entre agentes e outros modelos computacionais. Mais detalhes sobre estas diferenças podem ser encontrados em (FERBER, 1995; WOOLDRIDGE, 2002).

2.1.7.1 Agentes e sistemas especialistas

Sistemas especialistas (SE) são a mais importante tecnologia de IA produzida na década de 80 (HAYES-ROTH; WATERMAN; LENAT, 1983). São sistemas capazes

 $^{^6{\}rm Embora}$ reativa, a "subsumption architecture" é uma arquitetura deste tipo.

de solucionar problemas ou dar sugestões em domínios ricos em conhecimento. Agem prioritariamente como consultores de um usuário.

As 3 (três) principais diferenças entre agentes e SE são as seguintes (WOOL-DRIDGE, 2002):

- 1. SE não agem num ambiente, e sim através de um usuário utilizado como meio de sua ação;
- 2. SE não são usualmente capazes de apresentar um comportamento reativo ou pró-ativo;
- 3. SE não têm usualmente habilidades sociais que lhes permitam interagir com outros SE.

Pode-se também perceber que um SE não poderia ser considerado autônomo em relação ao seu ambiente, nem em relação às suas motivações, conforme os conceitos apresentadas na seção 2.1.4.

2.1.7.2 Agentes e módulos

As 2 (duas) principais diferenças entre agentes e módulos são as seguintes (FER-BER, 1995):

- 1. módulos não tem objetivos a atingir nem funções de satisfação a otimizar;
- 2. eventuais mecanismos de interação entre módulos são de baixo nível, como chamada de procedimentos.

Torna-se fácil perceber ainda que a noção de módulo não pode ser associada a nenhum dos significados de autonomia definidos na seção 2.1.4.

2.1.7.3 Agentes e objetos

Objetos são modelos computacionais que encapsulam dados e comportamentos. Caso tenham uma estrutura semelhante, podem ser agrupados em classes, as quais podem ser especializadas e generalizadas através de um mecanismo de herança. Apesar de algumas semelhanças, existem pelo menos 2 (duas) grandes diferenças entre tais modelos (FERBER, 1995; WOOLDRIDGE, 2002):

- objetos são definidos por um certo número de serviços (seus métodos), e caso recebam uma mensagem de outro agente solicitando a sua ativação, não podem recusar-se a fazê-lo. Além disto, objetos não são guiados por objetivos;
- 2. enquanto objetos executam métodos gratuitamente, agentes executam ações porque o desejam. Objetos não apresentam as propriedades de pró-atividade, reatividade e habilidade social. Finalmente, agentes mantém seu próprio "thread" de controle, enquanto objetos geralmente compartilham o mesmo "thread" de controle.

Relevância: Neste trabalho, o modelo de objetos será considerado como um modelo de implementação adequado para a codificação de agentes. Considerase assim que ambos os modelos representam fenômenos em níveis distintos de descrição de sistemas, como mostrado na seção 2.3.2.

2.2 Sistemas multiagentes

Nesta seção, são apresentados os principais conceitos envolvidos no estudo de SMA, compreendendo uma tentativa de definição da área, seu histórico, uma taxonomia de classificação de tais sistemas, os domínios de aplicação mais adequados e uma comparação com outros modelos computacionais.

2.2.1 Definição

Assim como no caso de agentes, não existe ainda uma definição com semântica precisa para SMA. Tal fato é melhor explicado quando se observa que este domínio utiliza resultados, conceitos e práticas de disciplinas tão díspares quanto IA, sociologia, economia, ciências da organização, administração e filosofia.

Entretanto, mais recentemente tem havido uma convergência de opiniões sobre o tema, que pode ser ilustrada através do seguinte texto, expresso por Weiss (1999, pag. 1), no prólogo de seu livro:

O objetivo de longo termo da IAD é o de desenvolver mecanismos e métodos que permitam aos agentes interagir entre si tão bem (ou melhor) quanto os humanos, assim como o de compreender as interações entre entidades inteligentes, sejam humanas, computacionais ou ambas. Este objetivo gera um certo número de desafios, todos ao redor da questão fundamental de quem interage com quem e quando.

Para que tal definição possa ser mais abrangente, e mais adequada a este trabalho, propõe-se aqui alterá-la do seguinte modo:

O objetivo de longo termo dos SMA é consistir na dimensão computacional de uma possível ciência da interação, provendo mecanismos de análise e síntese de sistemas destinados a esclarecer e a explorar a questão fundamental de quem interage com quem, quando, sobre o que, como e porque.

Tomando como exemplo o cenário de interação de robôs, caso Millie se convença dos argumentos apresentados por Clotaire e Berthold, na próxima vez que tiver de ser deslocada em condições desfavoráveis (quando) irá solicitar através de uma mensagem (como) a William (com quem) que a transporte (sobre o que), já que esta é sua função (porque) dentro da oficina.

De acordo com esta perspectiva, as **interações** e as **organizações** dos agentes são conceitos fundamentais para o avanço do domínio, e por isto são tratadas com mais detalhes respectivamente nos capítulos 3 e 4. Além disso, tais conceitos são também utilizados na seção 2.3.3 quando uma proposta de classificação para as pesquisas na área é apresentada.

2.2.2 Taxonomia de sistemas multiagentes

Garcia e Sichman (2003) propuseram uma taxonomia na qual um SMA pode ser classificado segundo alguns eixos⁷ tais como:

 $^{^7 \}rm Novamente, tal como na seção 2.1.2, apenas os eixos relevantes para este trabalho são aqui apresentados.$

- Eixo de Abertura: um SMA pode ser aberto ou não, indicando a possibilidade de o sistema alterar a sua composição dinamicamente, alterando o número de agentes que o compõem;
- Eixo de Granularidade: tipicamente, um SMA pode conter poucos agentes (SMA com baixa granularidade) ou milhares de agentes (SMA com alta granularidade);
- Eixo de Composição: um SMA pode ter agentes homogêneos (por exemplo, agentes idênticos podem tratar diferentes porções do espaço de entrada) ou heterogêneos (por exemplo, agentes têm tratamentos complementares para solucionar um determinado problema).

Esta taxonomia encontra-se representada de forma resumida na tabela 2.2.

Eixo	Tipos de Sistemas Multiagentes							
abertura	aberto				fechado			
composição	homogêneo		heterogêneo		homogêneo		heterogêneo	
granularidade	alta	baixa	alta	baixa	alta	baixa	alta	baixa

Tabela 2.2: Taxonomia de sistemas multiagentes (GARCIA; SICHMAN, 2003)

Nas partes II e III, são analisados prioritariamente os SMA abertos, de baixa granularidade e compostos por agentes homogêneos ou heterogêneos⁸.

2.2.3 HISTÓRICO

Segundo Sichman (1995), as primeiras tentativas para resolver problemas de modo cooperativo ocorreram no final da década de 1970 (FERBER; GASSER, 1991). Uma delas foi o projeto HEARSAY-II (ERMAN et al., 1980), um sistema para o reconhecimento de fala que originou o modelo de resolução de problemas conhecido como quadro-negro ("blackboard") (ENGELMORE; MORGAN, 1988). Nesta mesma época, surgiriam alguns modelos computacionais como os "beings" (LENAT, 1975) e os atores (AGHA; HEWITT, 1986), desenvolvidos com o objetivo de

 $^{^8{\}rm A}$ única suposição de homogeneidade neste caso se refere à presença do mesmo mecanismo de raciocínio social ou organizacional.

tratar problemas relacionados com recursos compartilhados, estruturas de controle complexas e o aparecimento de comportamentos sofisticados a partir de interações muito simples.

No início da década de 1980, nota-se pela primeira vez a importância da representação explícita, no seio de um agente, do conhecimento sobre outros agentes (KONOLIGE; NILSSON, 1980), num contexto de planejamento. A utilização de modelos organizacionais humanos em sistemas computacionais foi discutida (FOX, 1981), e surgiu o modelo da Rede Contratual (RC) (SMITH, 1980), um protocolo de comunicação baseado na noção de mercado econômico. Na mesma época, as técnicas de orientação a objetos começaram a tornar-se populares (STEFIK; BOBROW, 1983). Também nesta época, surgem os chamados robôs reativos de Brooks (1986), baseados na "subsumption architecture" descrita na seção 2.1.3. O conceito de sistema aberto também aparece pela primeira vez (HEWITT, 1986), fundamental para SMA. Finalmente, alguns ambientes computacionais foram desenvolvidos para testar e implementar essas idéias, como DVMT (CORKILL; LESSER, 1983), DPSK (CARDOZO, 1987) e MACE (GASSER; BRAGANZA; HERMAN, 1987).

2.2.4 Domínios de aplicação

Os SMA consistem em uma alternativa de solução interessante para problemas em domínios de aplicação particulares, que apresentem as seguintes características conforme discutido em Weiss (1999, pag. 7):

- domínios inerentemente distribuídos, onde os dados e informações a serem processados estão localizados em diferentes locais (distribuição geográfica), ou surgem em tempos distintos (distribuição temporal), ou ainda devem ser interpretados segundo ontologias e linguagens distintas (distribuição semântica) ou finalmente cujo acesso e uso requeiram capacidades perceptivas e cognitivas distintas (distribuição funcional);
- domínios *inerentemente complexos*, no sentido de serem grande demais para serem solucionados por um simples sistema centralizado, devido às limitações atuais da tecnologia de hardware ou de software.

Como exemplos de domínios que apresentam tais características, podem-se citar sistemas de comércio e mercados eletrônicos, gestão e monitoramento de redes de telecomunicações e de tráfego aéreo, otimização de logística de sistemas de transporte e otimização de processos de controle industriais, entre outros.

Uma descrição mais detalhada de aplicações da IAD e SMA na indústria pode ser encontrada em (PARUNAK, 1996; PARUNAK, 1999).

Relevância: Na parte III deste trabalho, são discutidas algumas aplicações nos domínios de processamento de linguagem natural (projeto \mathcal{N} ALAMAS), robótica (\mathcal{J} OJTEAM), otimização de processos industriais (\mathcal{O} PTIMA) e busca de informações na Internet (\mathcal{C} OMMUNET).

2.2.5 Sistemas multiagentes e outros modelos computacionais

Para finalizar esta seção, discutem-se abaixo as principais diferenças entre SMA e outros modelos computacionais. Mais detalhes sobre estas diferenças podem ser encontrados em (FERBER, 1995; WOOLDRIDGE, 2002; HÜBNER; SICHMAN; BECERRA, 2000).

2.2.5.1 Sistemas multiagentes e algoritmos paralelos para IA

Segundo Ferber (1995), os mesmos argumentos utilizados na seção 2.1.7 para diferenciar agentes de módulos podem ser empregados para distinguir SMA e algoritmos paralelos para IA: seus elementos constituintes não tem objetivos a atingir, e seus mecanismos de interação são elementares, como por exemplo o envio de sinais de sincronismo.

2.2.5.2 Sistemas multiagentes e objetos ativos

O conceito de *objeto ativo* é uma generalização do conceito de objeto (CARDOZO; SICHMAN; DEMAZEAU, 1993). Em linguagens de programação orientadas a objetos típicas, classes são tipos de dados e objetos são dados armazenados em memória, alocados de modo estático (em tempo de compilação) ou dinâmico (em tempo de execução). Objetos não têm um "thread" de controle associado, sendo

que usualmente todos os objetos do sistema compartilham do mesmo "thread" de controle. Consequentemente, operações realizadas em diferentes objetos são realizadas de forma serial.

O modelo de objetos ativos generaliza as propriedades de objetos descritas acima, tendo as seguintes propriedades: (i) múltiplos "threads" de controle, denominados servidores, onde cada servidor é capaz de processar um certo número de métodos; (ii) um estado privado acessível apenas aos servidores do objeto; e (iii) um estado público acessível a servidores definidos em outros objetos.

Num sistema de programação que possibilite a utilização do modelo de objetos ativos, sempre que uma determinada classe é instanciada, diversos "threads" de controle são criados. Objetos podem ser distribuídos pela rede, e operações sobre objetos podem ser realizadas em paralelo. Métodos são invocados através de uma chamada remota de procedimentos, ou "Remote Procedure Call" (RPC), sendo automaticamente direcionadas pelo sistema de gerenciamento de objetos ao servidor que possa executar o método solicitado. O estado público do objeto consiste num conjunto de dados disponível através da rede, enquanto que o estado privado reside num espaço de endereçamento acessível somente aos servidores do objeto.

Pode-se constatar que a definição acima não engloba nenhum significado de autonomia, dentre aqueles apresentados na seção 2.1.4. Este fato pode ser ilustrado pela seguinte afirmação de Wooldridge (2002, pag. 27):

Objetos ativos são essencialmente agentes que não tem a habilidade de exibir um comportamento autônomo flexível.

2.2.5.3 Sistemas multiagentes e sistemas de objetos distribuídos

A engenharia de software tem avançado bastante nos últimos anos na definição de metodologias e técnicas para o desenvolvimento de sistemas em ambientes distribuídos. Um dos resultados é um alto nível de transparência dos aspectos de distribuição, alcançado pela elaboração do conceito de sistema de objetos distribuídos (SOD), utilizados no padrão CORBA (NICOL; WILKES; MANOLA, 1993).

Segundo as recomendações da International Standard Organization (ISO) para o desenvolvimento de sistemas abertos de processamento distribuído (ODP),

um sistema distribuído apresenta as seguintes características: distribuição geográfica, concorrência, inexistência de estado global, possibilidade de falhas parciais, comunicação assíncrona, heterogeneidade, autonomia⁹, evolução e mobilidade.

Como a especificação de um tal sistema não é trivial, sugere-se dividí-la em 5 (cinco) modelos definidos a partir de pontos de vista: empresa, informação, computação, engenharia e tecnologia. Uma visão mais detalhada destes pontos de vista pode ser encontrada em (HÜBNER; SICHMAN; BECERRA, 2000).

Sob o ponto de vista de engenharia, os objetos num SOD estão fisicamente associados a um recurso de processamento (um computador, por exemplo) denominado $n\acute{o}$. Cada nó é controlado por um $n\'{u}cleo$ (um sistema operacional, por exemplo) que é responsável pela sua inicialização, pela criação de grupos de objetos, pelo suporte à comunicação, etc. Cápsulas são entidades (por exemplo, um processo) que possuem memória e que compartilham o nó. Numa cápsula, há um objeto especial, denomiado gerenciador, que a controla. Objetos são normalmente agrupados em cápsulas para facilitar a sua comunicação. Cada cápsula pode conter vários objetos formando agrupamentos. Os objetos de um mesmo agrupamento formam uma unidade que pode ser transferida, desativada, reativada e movida para outro nó. A comunicação dos objetos dentro do agrupamento é muito eficiente, já que os objetos são criados conjuntamente e desenvolvidos na mesma linguagem, e portanto sua comunicação normalmente é constituída de uma invocação de método. Assim como o nó e a cápsula, cada agrupamento é controlado por um gerenciador. Para que objetos de diferentes agrupamentos possam se comunicar, sugere-se uma estrutura em camadas que não será aqui detalhada, e que pode ser consultada em (HÜBNER; SICHMAN; BECERRA, 2000).

Tomando-se o ponto de vista de Engenharia apresentado acima, um SMA pode ser visto como um conjunto de cápsulas distribuídas em vários nós, sendo um agente representado por um agrupamento de objetos. Neste contexto, os gerenciadores têm suas funções especializadas. O gerenciador do nó tem a função de criar SMA, e o gerenciador de cápsula tem a função de criar, desativar, mover, etc. agentes para o sistema que representa. Já o gerenciador de agrupamento passa a fazer parte da estrutura interna dos agentes, criando os objetos necessários

⁹Esta noção corresponde claramente ao primeiro significado de autonomia discutido na seção 2.1.4.

ao seu funcionamento.

Relevância: Para concluir, observa-se que existem semelhanças entre as duas últimas comparações e aquela apresentada na seção 2.1.7, referente às diferenças entre agentes e objetos. Neste trabalho, o modelo de objetos ativos, bem como o de um SOD, será considerado como um possível modelo de implementação adequado para SMA. Em particular, o ambiente MASENV, descrito na seção 7.1, segue o modelo de agentes ativos. Já o ambiente SACI, descrito na seção 7.2, é inspirado no modelo de SOD. Assim, considera-se que SMA, SOD e objetos ativos representam fenômenos em níveis distintos de descrição de sistemas, como mostrado na seção 2.3.2.

2.3 Classificação de pesquisas em sistemas multiagentes

Para encerrar este capítulo, apresenta-se a seguir uma proposta para classificar pesquisas em SMA. Mais genérica e mais abstrata que as taxonomias apresentadas nas seções 2.1.2 e 2.2.2, tal proposta visa melhor situar qual é a exata contribuição dos diversos trabalhos realizados na área. Particularmente, ela será utilizada para classificar os modelos e aplicações desenvolvidas pelo autor, por seus alunos e por seus colaboradores, apresentados respectivamente nas partes II e III deste trabalho.

A proposta é estruturada através de 3 (três) eixos básicos: a finalidade da pesquisa, o nível de descrição utilizado para caracterizá-la e a dimensão específica que ela aborda. Tais eixos, independentes entre si, são descritos a seguir.

2.3.1 Resolução social e simulação social

No seio da comunidade científica da IA, já ocorre há algum tempo um debate referente aos objetivos científicos da área. Podem-se distinguir duas perspectivas relativamente opostas (FARRENY; GHALLAB, 1987): perspectiva de simulação e perspectiva de resolução. Segundo a perspectiva de simulação, a IA é um ramo da ciência cognitiva, cujo objeto de estudo é o raciocínio. Entretanto, diferentemente de outros ramos desta ciência, a IA verifica e testa seus modelos e suas

teorias não por meio de experimentações em seres humanos, mas através da programação em computadores. Já a perspectiva de resolução entende que a IA é um ramo da informática e engenharia, no qual o computador e seus programas não são meras ferramentas de pesquisa, mas os próprios objetos desta pesquisa. A IA busca a concepção de máquinas e métodos computacionais associados que realizem funções que necessitam de inteligência para serem desempenhadas. De um certo modo, os pesquisadores de IA fazem "Engenharia de Máquinas Inteligentes".

Este mesmo debate referente à dualidade ciência cognitiva/informática da IA também ocorre no seio da comunidade de SMA, quanto à finalidade do domínio (CASTELFRANCHI, 1990):

- Simulação Social: IAD/SMA é um ramo da ciência cognitiva, cujo objeto de estudo são as interações sociais, e cujo objetivo é formalizar, explicar e analisar por quê os agentes interagem, por quê cooperam e quais objetos mentais devem representar tais fenômenos;
- Resolução Social: IAD/SMA é um ramo da informática e engenharia, cujo objetivo é desenvolver técnicas cuja base seja formada por múltiplos agentes, suas interações e suas organizações, para resolver problemas de forma cooperativa e distribuída.

De acordo com a segunda visão apresentada acima, os pesquisadores de IAD/SMA de um certo modo fazem "Engenharia de Sociedades de Máquinas Inteligentes".

Tais perspectivas, embora filosoficamente disjuntas, não são mutuamente exclusivas, em alguns aspectos são até complementares. Pode-se assim construir um círculo virtuoso entre estas duas finalidades que a princípio parecem disjuntas, mas na realidade se realimentam dos resultados uma da outra para poder avançar o conhecimento nas áreas de IA e IAD/SMA.

2.3.2 Teorias, arquiteturas e linguagens de agentes

Wooldridge e Jennings (1995b) identificaram 3 (três) níveis de descrição presentes no estudo e desenvolvimento de SMA:

- teorias: o que são exatamente agentes? Que propriedades devem ter, e como se pode representá-las e raciocinar formalmente sobre elas?
- arquiteturas: como se podem construir agentes que satisfaçam as propriedades que se esperam de sua atuação? Quais as estruturas de harwdware e de software apropriadas para esta tarefa?
- linguagens: como agentes devem ser programados? Quais as primitivas adequadas para esta tarefa? Como se pode compilar e executar programas baseados em agentes de modo eficiente?

O interesse em se estabelecer modelos formais precisos para descrever agentes é a possibilidade que estes nos oferecem em prever o comportamento destes agentes numa determinada situação¹⁰. Normalmente, utilizam-se descrições lógicas (modais, de ações, temporais, de crenças) para precisar o comportamento dos agentes (WEISS, 1999, cap. 8) (O'HARE; JENNINGS, 1996, cap. 10).

Quanto ao aspecto das linguagens, alguns autores como Shoham (1991) acreditam que o paradigma de agentes já é maduro o suficiente para poder propor um novo paradigma de linguagens de programação. Esta não é a opinião da maioria dos pesquisadores da área, já que vários sistemas e aplicações utilizaram técnicas de objetos ativos ou distribuídos, como discutido na seção 2.2.5. Finalmente, cabe ressaltar ainda que alguns métodos de concepção de sistemas orientados a agentes têm surgido nos últimos anos, tais como GAIA, Agent UML e DESIRE, para facilitar a concepção de SMA. Uma ótima introdução a estes métodos se encontra em (WOOLDRIDGE, 2002, pag. 226-233).

2.3.3 Agentes, ambientes, interações e organizações

Em (DEMAZEAU, 1995), é apresentada uma metodologia para o desenvolvimento de SMA, ilustrada na figura 2.3. Segundo esta metodologia, uma aplicação poderia ser desenvolvida através da utilização de elementos básicos pré-existentes, disponíveis numa plataforma de desenvolvimento, de modo semelhante à utilização de um "shell" no processo de construção de um SE. Estes elementos básicos, por sua vez, utilizariam um sistema distribuído subjacente, de modo a tratarem exclusivamente a dimensão multiagentes.

¹⁰Semelhantemente à utilização de modelos formais em Engenharia de Software, por exemplo.

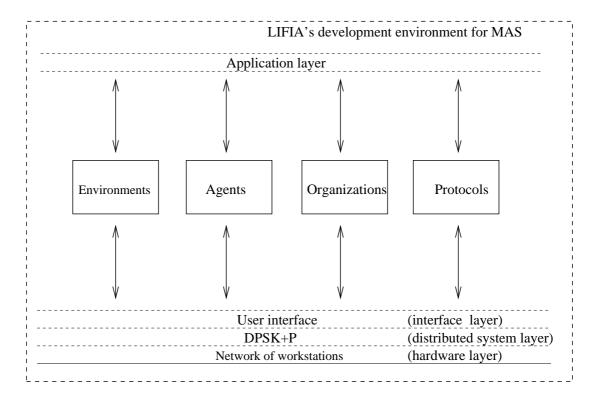


Figura 2.3: Metodologia Vogais (CARDOZO; SICHMAN; DEMAZEAU, 1993).

Esta metodologia adota como hipótese que um SMA teria uma funcionalidade global maior do que a simples soma das funcionalidades de seus agentes constituintes. Tal diferença se deveria ao fato da ação coletiva dos agentes, capturada através de suas interações e de suas organizações, propiciar um acréscimo potencial na funcionalidade do sistema global. Por outro lado, a metodologia adota também um princípio de recursividade: um SMA pode ser considerado como um único agente, se analisado num maior nível de abstração. Como exemplo desta idéia, pode-se considerar que um contrato fechado entre duas companhias coloca em interação dois agentes, mesmo se cada um deles for composto por centenas de pessoas.

Tal metodologia foi posteriormente denominada de Vogais, por considerar que o desenvolvimento de um SMA deveria se compor através do desenvolvimento de 4 (quatro) dimensões distintas:

• agentes: caracterizam os elementos básicos ativos do sistema, conforme descritos na seção 2.1;

- ambientes¹¹: caracterizam os elementos passivos do sistema;
- protocolos de interação: caracterizam os meios pelos quais os agentes trocam informações e controle entre eles;
- organizações: caracterizam as políticas a serem seguidas para restringir as interações entre os agentes.

Assim, a idéia básica da metodologia é que estas dimensões poderiam ser projetadas e implementadas independentemente, e depois integradas conforme a aplicação assim o necessitasse. Por exemplo, uma aplicação composta por agentes do tipo BDI, como os descritos na seção 2.1.6, que utilizassem um modelo de organização hierárquica e que interagissem por meio de um protocolo do tipo RC¹², poderia reutilizar, e eventualmente especializar, uma parcela do código já disponível na plataforma.

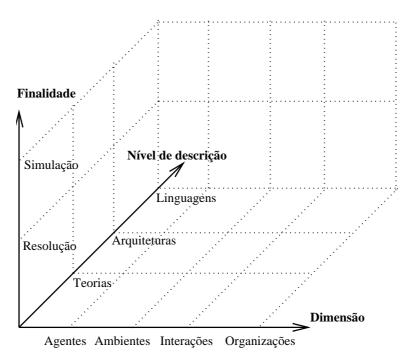


Figura 2.4: Visão integrada da classificação de pesquisas em sistemas multiagentes.

A figura 2.4 apresenta uma visão integrada dos três eixos da proposta de classificação, representados como eixos de um espaço tridimensional.

¹¹Do inglês *Environment*, o que explica o nome da metodologia (AEIO).

 $^{^{12}\}mathrm{Tal}$ protocolo é descrito com mais detalhes na seção 3.5.3.

2.4 Conclusões 33

2.4 Conclusões

Este capítulo apresentou os principais conceitos relacionados com as noções de agentes e SMA. Em particular, no restante do manuscrito o interesse deste trabalho é direcionado para agentes autônomos, cognitivos, sociais, que atuam em rede ou em máquinas isoladas, e que constituem um SMA aberto. No nível de implementação, são utilizados os modelos de objetos, objetos ativos e SOD para implementar tais sistemas.

Nos capítulos 3 e 4, que encerram a parte I deste manuscrito, são discutidas as dimensões relativas às *interações* e às *organizações* dos agentes, vitais para a compreensão dos trabalhos apresentados nas partes II e III.

3 Interações entre Agentes

O objetivo deste capítulo é fornecer ao leitor uma visão abrangente das interações entre agentes. Inicialmente, apresenta-se na seção 3.1 a noção de interferência social. Esta noção serve como ponto de partida para caracterizar possíveis situações de interação entre agentes, cuja classificação é discutida na seção 3.2. A seguir, na seção 3.3, discute-se ainda a representação da interação, indicando a importância dos próprios agentes serem capazes de detectar uma situação de interação potencial. Se assim o forem, estão presentes as condições necessárias para a ocorrência de uma resolução cooperativa de problemas, cujas fases são descritas na seção 3.4. Uma vez caracterizada uma situação de interação, descritos na seção 3.5. Na seção 3.6, discutem-se alguns aspectos relacionados à comunicação entre agentes, notadamente os chamados protocolos de comunicação, que são destinados a representar de forma mais eficiente o fluxo de comunicação entre agentes que ocorre em alguns dos métodos de interação. Finalmente, a seção 3.7 apresenta as conclusões deste capítulo.

3.1 Interferência social

Suponha que dois agentes ag_i e ag_j co-existam num mesmo ambiente e que possuam objetivos próprios. O efeito da ação individual de ag_i na obtenção dos objetivos de ag_j se denomina interferência social (de ag_i sobre ag_j) (CONTE; CASTELFRANCHI, 1992).

Esta interferência pode ser positiva (caso a ação de ag_i aproxime ag_j de seus objetivos), negativa (caso a ação de ag_i afaste ag_j de seus objetivos) ou neutra.

Como ag_j também realiza ações, pode ser interessante analisar os efeitos

mútuos das ações dos dois agentes, ou seja, a combinação da interferência de ag_i sobre ag_j com a de ag_j sobre ag_i . Esse fenômeno é chamado de interação social, sendo alguns padrões descritos na seqüência.

3.2 SITUAÇÕES DE INTERAÇÃO

Seja a seguinte definição, proposta por Ferber (1995):

Uma situação de interação é um conjunto de comportamentos resultantes de um grupo de agentes que agem para satisfazer seus objetivos, e que levam em conta as restrições devidas à limitação de recursos e à limitação de suas competências individuais.

Considere-se então que, para atingir seus objetivos, agentes realizem tarefas, que podem eventualmente utilizar recursos, consumíveis ou não, que se encontram disponíveis no ambiente. Tais situações de interação podem ser classificadas de acordo com 3 (três) dimensões distintas:

- 1. compatibilidade de objetivos: os objetivos dos agentes são considerados compatíveis/incompatíveis quando o fato de atingir um deles não acarretar/acarretar necessariamente a impossibilidade de atingir o outro;
- 2. quantidade de recursos: os recursos são considerados suficientes/insuficientes quando os agentes puderem/não puderem realizar suas tarefas simultaneamente;
- 3. competência dos agentes: a competência de um agente é considerada suficiente/insuficiente quando ele for capaz/incapaz de realizar sua tarefa, de modo a atingir seu objetivo¹.

Um exemplo ilustrativo no cenário de interação de robôs destas noções de objetivo, tarefa e recurso é a situação na qual Clotaire esteja soldando um peça (tarefa) para terminar a montagem da perna articulada de um futuro colega (objetivo), utilizando para isto um fio de solda (recurso).

¹O termo competente é aqui utilizado como sinônimo do terceiro significado de autonomia apresentado na seção 2.1.4.

A tabela 3.1 apresenta as possíveis situações de interação segundo tais dimensões, conforme (FERBER, 1995). Tais situações são detalhadas na seqüência.

Objetivos	Recursos	Competências	Situação de	Interferência	
			$Interaç\~ao$	Social	
compatíveis	suficientes	suficientes	$independ \hat{e}ncia$	neutra	
compatíveis	suficientes	insuficientes	$colabora$ ç $ ilde{a}o$	positiva	
			simples		
compatíveis	insuficientes	suficientes	$obstru$ ç $ ilde{a}o$	positiva	
compatíveis	insuficientes	insuficientes	$colabora$ ç $ ilde{a}o$	positiva	
			coordenada		
incompatíveis	suficientes	insuficientes	$competi$ ç $ ilde{a}o$	negativa	
			$individual\ pura$		
incompatíveis	suficientes	insuficientes	$competi$ ç $ ilde{a}o$	$\operatorname{negativa}$	
			$coletiva\ pura$		
incompatíveis	insuficientes	suficientes	conflito individual	negativa	
			por recursos		
incompatíveis	insuficientes	insuficientes	$conflito\ coletivo$	$\operatorname{negativa}$	
			por recursos		

Tabela 3.1: Situações de interação (FERBER, 1995)

3.2.1 Interferência social neutra

Dentre as situações descritas na tabela 3.1, apenas aquela denominada independência corresponde a uma interferência social neutra.

3.2.1.1 Independência

Nesta situação, não existe efetivamente uma real interação entre os agentes: cada qual realiza as suas ações de modo a atingir seus objetivos. Esta situação é chamada de cohabitação em (DEMAZEAU; MÜLLER, 1990).

3.2.2 Interferência social positiva

Dentre as situações descritas na tabela 3.1, existem 3 (três) que correspondem a uma interferência social positiva.

3.2.2.1 Colaboração simples

Esta situação consiste numa simples adição de competências, sem a necessidade de ações específicas de coordenação entre os componentes. A interação se exprime tipicamente através da divisão de tarefas e pelo compartilhamento de conhecimento. Os chamados SE "multi-experts", onde intervêm múltiplos especialistas de diferentes domínios, são um exemplo desta situação.

3.2.2.2 Obstrução

Esta situação caracteriza os cenários onde os agentes se atrapalham ao tentar atingir seus objetivos, mesmo que não necessitem do auxílio uns dos outros. Tal fato é conseqüência da quantidade escassa de recursos. Caso os recursos não sejam consumíveis, técnicas de coordenação de ações são necessárias para solucionar este tipo de interação. Exemplos típicos desta situação são congestionamento de veículos e problemas de alocação ótima de tarefas num processador, entre outros.

3.2.2.3 COLABORAÇÃO COORDENADA

Este tipo de situação é mais complexo, pois além de necessitarem da competência mútua, os agentes devem se coordenar para utilizar de modo apropriado os recursos escassos. Assim, tanto técnicas de alocação de tarefas como a coordenação de ações devem ser utilizadas ao mesmo tempo. O controle de redes, fabricação de produtos industriais e times de robôs móveis autônomos encontram-se tipicamente neste tipo de situação.

3.2.3 Interferência social negativa

Existem 4 (quatro) situações de interação descritas na tabela 3.1 que correspondem a uma interferência social negativa.

3.2.3.1 Competição individual pura

Quando seus objetivos são incompatíveis, os agentes devem negociar ou lutar para atingí-los. Caso não haja uma negociação, não existe verdadeiramente uma

interação entre os agentes: tipicamente "ganha o melhor". Um corrida esportiva individual é um exemplo deste tipo de situação.

3.2.3.2 Competição coletiva pura

Nesta situação, inicialmente os agentes se agrupam em torno de associações para poder atingir seus objetivos. Sua interação se dá internamente como uma colaboração coordenada. Num segundo momento, os grupos de opõem uns aos outros. Um exemplo típico desta situação são jogos esportivos em equipe.

3.2.3.3 Conflito individual por recursos

Por não serem suficientes, os recursos neste caso tornam-se a principal fonte de conflito entre os agentes. Exemplos desta situação são a posse de territórios no mundo animal ou a competição por uma vaga de trabalho numa empresa.

3.2.3.4 Conflito coletivo por recursos

Esta situação combina as duas anteriores, pois existe uma competição coletiva pura, aliada à escassez de recursos. Exemplos típicos são agrupamentos que lutam uns contra os outros pela posse de um território (guerra) ou de um mercado econômico.

3.2.4 Dimensões complementares

Uma vez apresentadas todas as possíveis situações de interação, cabe aqui ressaltar que se poderia acrescentar uma dimensão adicional às anteriores: o beneficiário da interação. Tal dimensão se justifica já que as interferências sociais podem ter efeitos distintos para os agentes: podem ser positivas para um deles, para todos eles, ou até mesmo neutras. Numa metáfora biológica, como a apresentada em (GARCIA; SICHMAN, 2003), a atuação de um vírus de computador poderia ser considerada como positiva para um dos agentes (vírus) e negativa para o outro (hospedeiro), caracterizando uma interação do tipo parasitismo.

Para concluir esta seção, cabe ainda citar aqui a existência de uma outra proposta de classificação das interações entre agentes, baseada em fundamen-

tos econômicos, e mais particularmente na Teoria dos Jogos (AXELROD, 1984). Um excelente resumo destes trabalhos pode ser encontrado em (WOOLDRIDGE, 2002).

Relevância: Nos trabalhos apresentados nas partes II e III, são exploradas apenas as situações de interação cuja interferência social seja positiva.

3.3 Representação das interações entre agentes

É importante ressaltar que a interferência social e as situações de interação descritas anteriormente existem como fatos objetivos, ou seja, ocorrem na realidade mesmo que os agentes envolvidos não estejam conscientes destes fatos.

Consequentemente, dependendo da presença ou da ausência de representação destes fenômenos pelos agentes, existem dois modos de caracterizar a ocorrência de interações entre agentes, descritos a seguir.

3.3.1 Interação intencional

Considera-se, neste caso, que a interação pode ser caracterizada por uma atitu-de (postura) dos agentes. No caso de uma situação de colaboração simples, se poderia afirmar, baseado em (FERBER, 1995), que:

Os agentes colaboram caso tenham firmado um compromisso comum de ação conjunta, após terem identificado e adotado um objetivo comum.

Em (SICHMAN, 1995), mostra-se que se os agentes estiverem conscientes de sua interferência social, eles podem raciocinar socialmente para tirar proveito da existência dos outros. Em particular, Conte e Castelfranchi (1992) mostram que agentes podem decidir tomar ações sociais, que têm como objetivo utilizar as ações de outros agentes ou tentar alterar seus estados mentais. Tais ações sociais serão melhor caracterizadas na seção 5.1.1.

3.3.2 Interação não intencional

A caracterização da interação no caso anterior apresenta 2 (dois) inconvenientes:

- retomando o cenário de interação de robôs, e imaginando que Berthold escorregue fazendo Millie cair ao ajudar Clotaire a carregá-la, ou então que deseje parar para recarregar sua bateria no caminho, será que existe de fato uma colaboração? Ou será que Millie não poderia ser mais eficaz ao realizar sua tarefa sozinha? Pela definição acima, a condição necessária e suficiente para ocorrência de colaboração é a presença de atitudes mentais adequadas, mesmo se o resultado da ação comum não for satisfatório;
- esta caracterização não poderia ser utilizada no caso de agentes reativos, pois como mostrado na seção 2.1.3 estes não têm uma representação interna explícita dos outros agentes, e no entanto cooperam entre si.

Assim, alguns autores consideram que as interações entre agentes podem ser qualificadas por um observador externo, quando este utiliza algum *índice* para medir esta atividade comum. Este é o caso, por exemplo, de uma colônia de formigas, onde observa-se claramente que se o número de formigas cresce, a quantidade de alimento recolhida ao ninho cresce numa proporção que só pode ser explicada pela existência de alguma ação coletiva.

Relevância: Os modelos apresentados na parte II podem ser utilizados em contextos onde a interação entre agentes seja intencional ou não intencional, e tal fato será ilustrado nas aplicações descritas na parte III.

3.4 Modelo de resolução cooperativa de problemas

Seja um conjunto de agentes cognitivos, que detectam entre si uma potencial situação de interação cuja interferência social seja positiva². Wooldridge e Jennings (1994) propõem um modelo de resolução cooperativa de problemas, composto por 4 (quatro) fases, que propicia a ação coletiva destes agentes:

²Apesar de não ser de interesse deste trabalho, o mesmo raciocínio pode ser empregado para a detecção de situações cuja interferência social seja negativa.

- 1. detecção do potencial de cooperação;
- 2. formação de coalizão;
- 3. formação de plano para a coalizão;
- 4. ação da coalizão.

Estas fases são resumidamente descritas na sequência.

3.4.1 Detecção do potencial de cooperação

Esta primeira fase se dá através do raciocínio social dos agentes, que inferem, por exemplo, que os objetivos e ações dos outros podem ser complementares aos seus. Normalmente, os agentes podem ter adquirido esta representação dos outros através de seus mecanismos de percepção e de comunicação, como ilustrado no modelo \mathcal{S}_{RM} e apresentado na seção 5.1.4.

3.4.2 FORMAÇÃO DE COALIZÃO

Uma vez detetado um potencial de cooperação, os agentes realizam um procedimento para formar uma *coalizão*. Por coalizão, entende-se uma organização dinâmica dos agentes, destinada a resolver um determinado problema.

3.4.3 Formação de plano para a coalizão

Quando uma coalizão é formada, os agentes devem então decidir como irão atingir seus objetivos. Trata-se de uma questão complexa, pois envolve distribuir tarefas entre os agentes e sincronizar a sua execução, de modo que ao seu final os objetivos sejam atingidos. Utilizam-se para tal atividade técnicas de planejamento em IA (RUSSEL; NORVIG, 1995, cap. 11), extendidas para levar em conta o aspecto multiagentes, que são melhor discutidas na seção 3.5.4. Em certos casos, esta tarefa é realizada durante a execução do sistema (planejamento "on-line"), enquanto que em outros casos ela é realizada antes da fase de execução (planejamento "off-line"), por exemplo especializando soluções de situações similares já tratadas no passado. Tal técnica, chamada raciocínio baseado em casos ("case-based-reasoning"), foi adotada no modelo $S_{\rm RM}$, descrito na seção 5.1.

3.4.4 AÇÃO DA COALIZÃO

Uma vez estabelecido um plano para nortear as ações dos agentes, a fase de resolução propriamente dita pode ser iniciada.

Uma vez que a ação coletiva chegue ao seu final, e os objetivos da coalizão forem atingidos, a coalizão é desfeita, e os agentes podem tomar a iniciativa de formar outras coalizões.

3.5 MÉTODOS DE INTERAÇÃO

Uma vez detetada uma situação de interação, deve-se prover os agentes de *métodos* de interação para poder concretizar esta situação. Admite-se aqui, de acordo com a definição de agente apresentada na seção 2.1, que os agentes têm capacidades de percepção, comunicação e ação. Tais métodos, inspirados em (FERBER, 1995), são brevemente descritos abaixo.

3.5.1 AGRUPAMENTO

No caso de SMA reativos, principalmente os inspirados no mundo animal, um método extramamente eficiente para possibilitar a interação é a simples aproximação física. Como conseqüência, os agentes tornam-se menos vulneráveis a ataques de predadores e ganham eficiência na tarefa de procura de alimento.

Um caso especial é a *multiplicação*, considerada aqui como o simples aumento quantitativo dos indivíduos da população. Além das vantagens apresentadas acima, o aumento dos agentes do sistema permite que este seja mais imune a falhas em seus componentes. Em (FERBER, 1995, pag. 88), mostra-se um exemplo interessante da utilização deste método em um sistema de robôs exploradores.

3.5.2 Especialização

Um outro método, também bastante utilizado em SMA reativos, é o processo de especialização que os agentes podem sofrer no decorrer de sua existência. Como um exemplo ilustrativo desta técnica, (DROGOUL, 1993) apresenta um estudo do comportamento de uma colônia de formigas, onde se observa que à medida

que a quantidade de alimento, de ovos ou de recém-nascidos se altera no ninho, a colônia ativa um mecanismo próprio de autoregulação, alterando a quantidade de formigas especializadas nestas tarefas. Inicialmente, todas as formigas têm a capacidade de realizar quaisquer destas tarefas, mas através de um mecanismo de reforço vão se tornando mais especializadas nas tarefas mais necessárias à colônia num determinado momento.

3.5.3 Divisão de tarefas e de resultados

Segundo Smith e Randall (1981), uma resolução distribuída de problemas por diversos agentes é realizada em 3 (três) fases:

- 1. decomposição do problema em subproblemas menores, sendo este procedimento repetido sucessivamente até que a granularidade dos subproblemas seja adequada para que um único agente o solucione;
- 2. resolução dos subproblemas, sendo que cada agente isoladamente resolve o subproblema que lhe foi alocado;
- 3. síntese da solução, quando se obtém a solução do problema original através da integração das soluções dos diversos subproblemas.

Para que este processo possa ocorrer, é necessário dividir tarefas, correspondentes aos subproblemas, entre os agentes do sistema. Adicionalmente, mesmo que as tarefas sejam independentes entre si, os agentes devem dividir resultados parciais, obtidos durante a fase de resolução de seus subproblemas.

Nos SMA cognitivos, a divisão de tarefas ocorre por meio de mecanismos de oferta e demanda. Existem mecanismos centralizados, onde um agente particular tem como função principal realizar a divisão de tarefas entre os demais agentes, visando otimizar a demanda em relação às ofertas recebidas.

Já nas abordagens decentralizadas, não existe esta figura do agente centralizador, sendo que todos os agentes do sistema podem requisitar e oferecer serviços. Basicamente, existem 2 (dois) dois tipos de técnicas que podem ser utilizadas:

1. técnicas informadas, onde os agentes têm uma representação das capacidades uns dos outros, bem como eventualmente do controle dos recursos do

ambiente;

2. técnicas não informadas, onde os agentes não têm esta representação, e usa-se um mecanismo do tipo "licitação", baseado no mercado econômico.

Um exemplo clássico de técnica não informada é a chamada Rede Contratual (SMITH, 1980). Nesta técnica, um agente contratante anuncia, através de uma mensagem difundida aos outros, que necessita que uma determinada tarefa seja executada sob determinadas condições, tais como custo máximo, tempo máximo de execução etc. Agentes que estejam dispostos a realizá-la enviam mensagens contendo ofertas. O agente contratante analisa então as ofertas recebidas, e escolhe a mais adequada segundo algum critério de elegibilidade, caso alguma delas preencha os requisitos necessários. Para concretizar o contrato, o agente contratante avisa ao contratado que enviou a proposta escolhida que a sua oferta foi aceita e um contrato estabelecido. Agentes podem assumir o papel de contratante e contratado indistintamente, conforme o processo de resolução assim o requeira³. Uma descrição simplificada da troca de mensagens entre os agentes para implementar esta técnica no cenário de interação de robôs é apresentada na seção 3.6.4.

O método de divisão de tarefas normalmente é utilizado na fase de formação de uma coalizão, descrita na seção 3.4.2. Na seção 5.1.11, mostra-se como o modelo \mathcal{S}_{RM} pode servir como base para um método decentralizado informado de divisão de tarefas, denominado de coalizões baseadas em dependência (CBD).

Ferber (1995, pag. 408-411) apresenta ainda uma técnica de alocação emergente de tarefas, baseada em comunicação indireta através de sinais no ambiente, que serve para explicar este fenômeno de alocação de tarefas em SMA reativos.

3.5.4 Coordenação de ações

Malone (1988, apud FERBER 1995) define a problemática de coordenação de ações da seguinte maneira:

As atividades de coordenação consistem em todas as atividades suplementares que devem ser cumpridas num ambiente multiagentes, e

³A RC propõe ainda outras trocas de mensagens, para acompanhar a ação da coalizão e a avaliação dos resultados desta ação, que não são aqui descritas.

que não seriam necessárias caso um único agente estivesse tentando alcançar isoladamente os mesmos objetivos.

Agentes podem coordenar suas ações basicamente através de sincronização, planejamento, coordenação reativa e regulamentação.

3.5.4.1 SINCRONIZAÇÃO DE AÇÕES

A sincronização de ações consiste no aspecto mais elementar de coordenação, que visa garantir o encadeamento dos ações realizadas pelos diversos agentes. Tal mecanismo é utilizado quando deve-se gerir a simultaneidade das ações e verificar a coerência do resultado final. Normalmente, utilizam-se técnicas de sistemas distribuídos para este fim.

Como um exemplo ilustrativo no cenário de interação de robôs, Clotaire e Berthold devem sincronizar suas ações no momento em que devem colocar Millie na maca para ser transportada.

3.5.4.2 Planejamento de ações

Sem dúvida alguma, o planejamento de ações é uma das técnicas mais tradicionais de IA (RUSSEL; NORVIG, 1995, cap. 11-13). Basicamente, trata-se de dividir o tratamento de ações em duas fases. Na primeira, raciocina-se sobre o conjunto de todas as ações que devem ser realizadas para atingir um determinado objetivo, e gera-se um ou mais planos ao seu final. Numa segunda fase, escolhe-se um dos planos gerados e passa-se a executá-lo. No caso de ambientes dinâmicos, torna-se necessário eventualmente replanejar dinamicamente as ações, ou até mesmo intercalar fases de planejamento e de ação.

No caso de SMA, tal tarefa é mais complexa, pois como visto na seção 3.1, existe uma interferência social entre os agentes, e seus planos individuais podem gerar conflitos de objetivos ou de acesso a recursos.

Durfee (1999) propõe uma taxonomia para planejamento em SMA, composta por 3 (três) categorias:

• planejamento centralizado, onde um único agente realiza a etapa de planejamento, cujo produto final consiste num único plano, que depois é distribuído

a todos os agentes do sistema para que realizem as ações a eles atribuídas neste plano;

- planejamento decentralizado de plano único, onde um grupo de agentes, tipicamente representando diferentes especialistas em aspectos complementares do domínio, trabalha de forma cooperativa para gerar um único plano, que depois é executado pelos agentes do sistema;
- planejamento decentralizado de planos distribuídos, onde um grupo de agentes, tipicamente interessados em seus próprios objetivos individuais, coopera para gerar vários planos individuais de ação, de modo a minimizar ou eliminar eventuais conflitos.

A última categoria apresentada acima é a mais complexa, pois na maior parte dos casos os agentes somente representam a porção limitada do plano que lhes interessa. Assim, não existe em local algum do sistema uma representação do plano multiagentes global, que envolve as ações de todos os agentes.

3.5.4.3 COORDENAÇÃO REATIVA

Esta técnica é aplicada aos SMA reativos, e utilizam basicamente o ciclo percepção-ação destes agentes, por exemplo através de modelos baseados em variações da Teoria de Campos de Forças, utilizada na Física. Tal modelo foi adotado numa aplicação do projeto $\mathcal{N}_{\text{ALAMAS}}$, descrita na seção 10.3.2, que utilizou uma abordagem reativa.

3.5.4.4 COORDENAÇÃO POR REGULAMENTAÇÃO

Destinada a eliminar eventuais conflitos a priori, antes que estes apareçam, esta técnica utiliza regras de "bom" comportamento a serem seguidas pelos agentes do sistema, como regras de prioridade.

Como um exemplo ilustrativo no cenário de interação de robôs, Clotaire e Berthold poderiam ter acesso à maca para transportar Millie, desde que outros robôs com maior prioridade no uso deste recurso não necessitassem utilizá-lo no mesmo instante.

Mais detalhes sobre a coordenação de agentes, notadamente sobre as duas últimas técnicas descritas acima, podem ser encontrados em (FERBER, 1995, cap. 8) e em (WOOLDRIDGE, 2002, cap. 9).

3.5.5 Resolução de conflitos

Retomando mais uma vez o cenário de interação de robôs, sejam as seguintes situações:

- 1. quando Clotaire solicita a Berthold que a ajude a carregar Millie de volta ao seu local de trabalho, este lhe responde que não será possível, já que nas próximas horas deve se ocupar, juntamente com William, a transportar Yun, um robô japonês que pesa espantosos 470 kg, e que é especialista na luta de sumô;
- 2. depois de finalizada sua tarefa, Berthold retorna extenuado mas aceita auxiliar Clotaire a transportar Millie; entretanto, ao contrário da opinião desta última, insiste que utilizem a maca transportadora para que, desta vez, não haja riscos de que a cabeça de Millie possa ser danificada, como ocorrido da última vez em que a transportou;
- 3. uma vez iniciado o transporte, Clotaire dá uma bronca em Berthold, que ao desempenhar sua ação de maneira brusca deslocou o cotovelo de Millie ao tentar virar seu braço esquerdo, no momento em que Clotaire se ocupava, com a paciência típica de um microsoldador, de virar o braço direito de Millie.

As situações acima têm algo em comum: a ocorrência de um conflito entre os agentes. Do mesmo modo que ocorre na sociedade humana, sempre que agentes autônomos agem coletivamente não existe nada mais comum do que a ocorrência de pontos de vista distintos em relação a uma questão.

Conflitos podem ocorrer nas diversas fases de uma resolução cooperativa de problemas. Em particular, nos cenários ilustrados acima, os conflitos ocorrem respectivamente na segunda (formação de coalizão), terceira (formação de plano para a coalizão) e quarta (ação da coalizão) fases do processo descrito na seção 3.4.

No caso dos SMA reativos, esta problemática não se coloca, pois a resolução dos conflitos (caso estes existam) já foi "internalizada" nos comportamentos préprogramados dos agentes.

Já no caso dos SMA cognitivos, deve-se prover meios para a resolução de tais conflitos, de modo a não colocar em risco a manutenção do sistema, e nem de prejudicar excessivamente seu desempenho. Tais meios são brevemente discutidos abaixo.

3.5.5.1 Arbitragem

O método mais simples para a resolução de conflitos é a escolha de um *árbitro*. Trata-se de uma solução centralizada, na qual escolhe-se um agente particular cuja função principal seja resolver os eventuais conflitos que possam ocorrem nas diversas fases da resolução cooperativa de problemas.

No segundo cenário de conflito de interação de robôs apresentado acima, um terceiro robô, por exemplo o chefe dos robôs transportadores, poderia impor que a maca fosse utilizada, caso avaliasse que Berthold, apesar de ocupar a função de transportador, não apresentasse habilidades suficientes para esta tarefa.

3.5.5.2 Negociação

Smith (1980, apud MÜLLER 1996) definem negociação do seguinte modo:

Por negociação, entende-se uma discussão na qual as partes interessadas trocam informações e chegam a um acordo. Neste processo, existem três componentes importantes: (i) a troca de informação é bilateral; (ii) cada uma das partes envolvidas avalia a informação através de sua própria perspectiva e (iii) o acordo final é obtido através de uma mútua escolha das partes.

Outra definição, proposta por Pruitt (1981, apud MÜLLER 1996), baseia-se em considerações psicológicas, e considera a existência de um conflito como ponto de partida de uma negociação:

Negociação é um processo através do qual uma decisão conjunta é

estabelecida entre duas ou mais partes. Tais partes expõem demandas contraditórias, e movem-se então em direção a um acordo, através de concessões e busca de novas alternativas.

Müller (1996) caracteriza os estudos e desenvolvimentos em negociação multiagentes através de 3 (três) categorias básicas:

- 1. linguagem de negociação, onde se busca definir as primitivas de comunicação adequadas para o processo de negociação, sua semântica, sua estruturação e sua utilização em protocolos de negociação;
- 2. modelos de decisão de negociação, onde se procura definir funções de utilidade e estruturas as preferências dos agentes, bem como suas estratégias de negociação;
- 3. processo de negociação, onde modelos globais do processo de negociação são investigados, bem como analisados os comportamentos dos participantes deste processo.

3.5.5.3 Votação

Uma maneira alternativa, também decentralizada, de resolução de conflitos se dá através de um processo de votação. Normalmente, um agente é responsável por detetar as opiniões conflitantes e sistematizar as possíveis soluções. Estas são então enviadas a todos ou a parte dos participantes, que votam na proposta que lhes parece mais adequada. Os votos podem ser paritários, ou então ponderados segundo a importância dos agentes na sociedade. Ao final, o agente que encaminhou a votação contabiliza o resultado e o envia aos participantes, que adotam a solução preferencial da maioria. Como exemplo, o trabalho apresentado em (FERREIRA; WAINER, 2000) mostra a utilização desta técnica, no caso de um SMA destinado ao agendamento automático de reuniões.

3.6 Comunicação entre agentes

A maior parte dos métodos de interação acima citados, tais como negociação, divisão de tarefas e resolução de conflitos, fazem uso extensivo da comunicação

entre agentes. Trata-se de um assunto de importância fundamental no estudo e desenvolvimento de SMA, e alguns de seus aspectos mais relevantes são descritos resumidamente a seguir.

3.6.1 Teoria dos atos de fala

A comunicação entre agentes cognitivos é baseada prioritariamente na chamada *Teoria dos Atos de Fala* ("Speech Act Theory"). Proposta inicialmente pelo filósofo John Austin (AUSTIN, 1962), no início da década de 60, foi posteriormente extendida pelo linguista John Searle (SEARLE, 1969), e inicialmente explorada em IA por Cohen e Perrault (1979, apud WOOLDRIDGE 2002).

Tal teoria, em resumo, considera primeiramente que o ato de enunciar uma frase em linguagem natural tem características similares às de uma ação física, pois ambas alteram o estado do mundo.

Esta teoria considera ainda que um ato de fala sempre contempla 3 (três) aspectos:

- 1. ato locucionário, referente ao ato de pronunciar a frase;
- 2. ato ilocucionário, referente à ação subjacente ao enunciado da frase;
- 3. ato perlocucionário, referente ao efeito da ação.

A título de exemplo, retomando uma vez mais o cenário de interação de robôs, podemos imaginar a seguinte frase, enunciada por Berthold, ao aceitar a proposição de Clotaire a carregar Millie de volta ao seu local de trabalho:

Puxa, Clotaire, estou tão cansado, fiz 23 transportes hoje e preciso recolocar óleo nas minhas juntas. Mas como posso negar um pedido seu? Afinal de contas, já está no final do turno e poderíamos ir trocar nosso óleo juntos depois, não é mesmo? Conheço um ótimo posto, novinho em folha, que abriu pertinho daqui, no mês passado

O ato ilocucionário nesta frase é a ação de sedução, e o ato perlocucionário correspondente, que Berthold deseja, é a aceitação do convite por parte de Clotaire. Observa-se assim que atos de fala, assim como ações físicas, podem ser

utilizados num mecanismo de planejamento. Neste exemplo, para atingir o objetivo de sair com Clotaire, Berthold utiliza a ação de seduzir, materializada pelo enunciado da frase mostrada acima.

Finalmente, atos de fala, assim como ações físicas, podem ser agrupados em categorias, correspondentes a seus conteúdos ilocucionários e perlocucionários, tais como (WOOLDRIDGE, 2002):

- representativos, que apresentam a opinião do emissor em relação ao estado do mundo, como por exemplo informar;
- diretivos, que expressam a vontade do emissor que o receptor realize determinada ação, como por exemplo requisitar;
- comissivos, que comprometem o emissor com uma ação, como por exemplo prometer;
- expressivos, que expressam algum estado psicológico, como por exemplo agradecer;
- declarativos, que efetuam mudanças num determinado estado institucional, como por exemplo promulgar uma lei.

3.6.2 Linguagens de comunicação

Baseadas na Teoria de Atos de Fala, foram desenvolvidas várias *linguagens* para suportar a comunicação entre agentes.

Inicialmente, cabe ressaltar que a metodologia Vogais (DEMAZEAU, 1995), apresentada na seção 2.3.3, propõe que uma linguagem de comunicação de agentes deva ter pelo menos 3 (três) camadas distintas:

- camada de endereçamento, relacionada com o sistema distribuído subjacente, onde exprimem-se a localização física dos agentes, o modo de transmissão da mensagem etc.;
- 2. camada multiagentes, também chamada de linguagem externa⁴, que trata

 $^{^4\}mathrm{Na}$ realidade, outros autores chamam de linguagem externa a combinação desta camada com a anterior.

especificamente dos aspectos multiagentes, como por exemplo expressando os atos ilocucionários de uma mensagem;

3. camada do domínio, também chamada de linguagem interna, que permite que se represente o conhecimento de um determinado domínio de discurso.

No início da década de 90, a agência de defesa americana (DARPA) financiou um projeto denominado "Knowledge Sharing Effort" (KSE), cujo objetivo era o de permitir a troca de conhecimento entre sistemas de informação autônomos, dotando-os assim de uma possível interoperabilidade. Como resultado deste projeto, foram propostas 2 (duas) linguagens, denominadas KIF ("Knowledge Interchange Format") e KQML ("Knowledge Query Manipulation Language") (FININ et al., 1994).

A linguagem KIF é uma linguagem interna, que permite que agentes expressem propriedades e relações entre os elementos do domínio. Trata-se de uma versão extendida do cálculo de predicados de primeira ordem, acrescida de funções aritméticas, cuja representação é similar à da linguagem LISP. Como exemplo, seja a seguinte expressão escrita em KIF:

(and(commit-task ?answer)(transport Millie room7 room4)(deadline 3))

No cenário de interação de robôs, esta frase poderia significar que dentro de um prazo de 3 (três) horas, é necessário transportar Millie da oficina 7 para a oficina 4, e se deseja uma resposta, afirmativa ou negativa, para um compromisso com esta ação.

A linguagem KQML é uma linguagem externa, que define formatos para mensagens a serem trocadas entre agentes. Cada mensagem pode ser considerada como um objeto, num sentido de linguagens de programação orientadas a objetos, pois contém uma performativa, que pode ser considerada como uma classe de mensagem e relacionada com o ato ilocucionário descrito na seção 3.6.1, e uma seqüência de parâmetros, que consistem em pares atributo-valor e que podem ser considerados como variáveis de instâncias. No cenário de interação de robôs, Clotaire poderia enviar a seguinte mensagem KQML a Berthold:

Nesta situação, Clotaire é o emissor da mensagem (sender) e deseja uma resposta (performativa ask-one) de Berthold, que é o receptor da mensagem (receiver). O conteúdo da mensagem (content) é a expressão na linguagem KIF (language) discutida anteriormente, utilizando-se os termos de uma ontologia (ontology) denominada ONT-ROBOT. A resposta deve fazer referência ainda ao código da mensagem enviada (reply-with).

Um outro exemplo de linguagem de comunicação de agentes, semelhante a KQML, é a linguagem ACL (Agent Communication language), proposta pela FI-PA (Foundations for Intelligent Physical Agents) (FIPA, 2001).

3.6.3 Ontologias de comunicação

Como visto no exemplo de comunicação entre os robôs mostrado acima, para que a interação entre agentes ocorra de modo efetivo, não basta que estes compartilhem uma mesma linguagem de comunicação, embora tal condição seja obviamente necessária. Os agentes precisam também estar de acordo sobre uma terminologia comum que utilizam para descrever o domínio. No exemplo, Berthold deve compreender que significado do termo (deadline 3) para Clotaire é que o transporte deve ser finalizado num período de 3 (três) horas a partir do anúncio. Tal requisito torna-se mais importante ainda em SMA abertos, cujos agentes não são projetados e/ou desenvolvidos por uma única equipe.

Uma especificação do significado dos termos utilizados num domínio, acrescida da relação entre estes termos, forma uma *ontologia*. O servidor **Ontolingua** (FARQUHAR; FIKES; RICE, 1997) é um servidor, acessível pela Internet que per-

mite que ontologias distintas escritas em KIF e produzidas por grupos diferentes possam ser compartilhadas. XML e DAML são exemplos de outras linguagens destinadas a descrever ontologias (XML, 2001; DAML, 2001).

3.6.4 Protocolos de comunicação

Uma vez definidas uma linguagem e uma ontologia comuns, agentes podem enviar mensagens uns aos outros. O encadeamento destas mensagens gera os chamados protocolos de comunicação. Retomando mais uma vez o cenário da interação de robôs, considere-se que Clotaire, ao invés de solicitar apenas a Berthold que a auxilie no transporte de Millie, tenha anteriormente enviado um pedido a todos os robôs da fábrica. Imagine-se ainda que além de Berthold, que se dispôs a ajudá-la imediatamente, William tenha lhe respondido negativamente e que Yun, o robô japonês, também lhe tenha oferecido seus préstimos, mas somente após ter terminado sua tarefa corrente, o que ocorreria em 30 minutos. Em vista destas respostas, Clotaire aceitaria a ajuda de Berthold, pois quer terminar esta tarefa de transporte o quanto antes.

Esta situação seria representada pela seguinte troca de mensagens em KQML⁵:

⁵Embora não representada neste exemplo, esta troca de mensagens poderia também conter uma resposta negativa à oferta de Yun.

```
: receiver Clotaire
   : in-reply-to ID293847
   : reply-with ID473829
   : language KIF
   : ontology ONT-ROBOT
)
(reply
   : content (commit-task no)
   : sender William
   : receiver Clotaire
   : in-reply-to ID293847
   : language KIF
   : ontology ONT-ROBOT
)
(reply
   : content (and (commit-task yes)
                   (start-time 0.5) )
   : sender Yun
   : receiver Clotaire
   : in-reply-to ID293847
   : reply-with ID382947
   : language KIF
   : ontology ONT-ROBOT
)
(reply
   : content (award yes)
   : sender Clotaire
   : receiver Berthold
   : in-reply-to ID473829
   : language KIF
   : ontology ONT-ROBOT
```

3.7 Conclusões 56

)

Este protocolo implementa, de modo simplificado, o método de divisão de tarefas por meio da técnica de RC, descrita na seção 3.5.3. O fato importante a ressaltar é que o projeto de protocolos de comunicação, como o descrito acima, favorece a reutilização dos métodos de interação, descritos na seção 3.5. Esta reutilização pode ocorrer, pois a seqüência de mensagens envolvida numa divisão de tarefas pela técnica da RC, ou uma segunda seqüência que implemente uma resolução de conflitos por votação, é independente do domínio de aplicação.

Relevância: Os ambientes \mathcal{M} ASENV e \mathcal{S} ACI, descritos respectivamente nas seções 7.1 e 7.2 deste manuscrito, implementam ambientes de comunicação entre agentes que possibilitam o desenvolvimento dos métodos de interação descritos neste capítulo. Em particular, o ambiente \mathcal{M} ASENV propicia uma interface para o desenvolvimento de protocolos de comunicação, segundo uma sintaxe própria, enquanto que o ambiente \mathcal{S} ACI utiliza a linguagem KQML como linguagem externa de comunicação entre seus agentes. No capítulo 8, as figuras 8.3 e 8.4 apresentam respectivamente os protocolos de formação de coalizão dos sistemas \mathcal{D} EPINT e \mathcal{D} EPINT +.

3.7 Conclusões

Neste capítulo observou-se um resultado importante:

As interações entre os agentes podem eventualmente criar organizações dinâmicas.

Tais organizações dinâmicas, aqui denominadas de coalizões, são melhor caracterizadas na seção 4.5.2, bem como na discussão do modelo \mathcal{S}_{RM} , apresentado na seção 5.1.

Caso padrões e métodos de interação envolvendo os mesmos agentes se repitam muitas vezes, eles podem ser capturados em estruturas pré-estabelecidas, diminuindo assim a complexidade inerente do processo de formação de coalizão e

3.7 Conclusões 57

dos métodos de divisão de tarefas e de coordenação. Consequentemente, formamse grupos mais perenes, que se mantém no tempo. Esta manutenção tem a vantagem de tornar o processo de resolução cooperativa de problemas mais eficiente, já que a fase de formação de coalizão é realizada a priori. Como desvantagem, estruturas rígidas tem uma maior dificuldade em se adaptarem a ambientes muitos dinâmicos. Esta discussão, a respeito das chamadas organizações de agentes, é apresentada no capítulo seguinte.

4 Organizações de Agentes

Este capítulo tem como objetivo fornecer ao leitor uma perspectiva geral das chamadas organizações de agentes. Inicialmente, discutem-se na seção 4.1 as diferentes definições às quais este termo pode estar associado. Algumas características destas organizações, comuns a quase todos os trabalhos na área, encontram-se descritas na seção 4.2. A finalidade de uma organização de agentes é então discutida na seção 4.3. Uma taxonomia para melhor caracterizar dimensões importantes destas organizações, tais como sua gênese, seu foco de interesse, seu nível de descrição e seu comportamento, é apresentada na seção 4.4, sendo as três primeiras posteriormente detalhadas respectivamente nas seções 4.5, 4.6 e 4.7. Para tornar possível caracterizar o fenômeno de reorganização, ligado à última dimensão e abordado na seção 4.9, discute-se previamente na seção 4.8 as vantagens obtidas pelos agentes em manter uma representação destes modelos organizacionais. Finalmente, a seção 4.10 apresenta as conclusões deste capítulo.

4.1 Definição

A noção de organização é facilmente percebida ao nosso redor. Por exemplo, o usuário de uma mesa de trabalho sabe se ela está bem organizada ou não, um grupo de onze pessoas jogando futebol é diferenciado de um grupo de onze pessoas numa multidão porque estão organizados, uma empresa é um sistema que somente existe como sendo organizado. A partir destes exemplos, pode-se inferir um primeiro propósito da organização dos sistemas: fazer o sistema alcançar facilmente seus objetivos (HÜBNER; SICHMAN, 2000b). Enquanto parece simples caracterizar um sistema como organizado baseando-se no modo como mantém e busca atingir seus objetivos, definir o que é uma organização, o que a constitui, quais suas formas e estruturas não é tarefa tão simples.

Curiosamente, Carley e Gasser (1999) argumentam que a resposta mais usual para a clássica pergunta "O que é uma organização?" é a seguinte: "Eu sei identificá-la quando a vejo". Segundo ainda o ponto de vista expresso por Boissier (1993) e por Garcia e Sichman (2003), a organização de um SMA pode ser definida do seguinte modo:

Uma organização de um SMA pode ser vista simplificadamente como um conjunto de restrições adotadas por um grupo de agentes para que possam atingir seus objetivos mais facilmente.

Um grupo de agentes pode adotar (ou ser submetido a) diferentes formas de organização e certamente esta opção terá conseqüências no desempenho do grupo. Entretanto, não há como pré-determinar se um projeto de organização trará um rendimento ótimo em qualquer circunstância. Vários fatores determinam o desempenho da organização de uma sociedade: o tipo das tarefas sendo realizadas pelos agentes, sua capacidade cognitiva, seu treinamento, a constância do ambiente, restrições legais ou políticas e o tipo de resultado esperado (eficiência, eficácia, custo mínimo, precisão etc.) (CARLEY; GASSER, 1999). A área de estudo que se dedica a este problema, denominada Teoria da Organização, é definida por Prietula, Carley e Gasser (1998) do seguinte modo:

A teoria organizacional pode ser caracterizada como o estudo de como múltiplos fatores influenciam o comportamento de organizações, pessoas e tecnologias que a compõem.

4.2 Características de organizações de agentes

Apesar de não haver uma única definição universalmente aceita de organizações de agentes, segundo Carley e Gasser (1999) existem, entretanto, algumas características básicas presentes em tais organizações sobre as quais existe um certo consenso na área:

• empregam tecnologias de resolução de problemas de larga escala;

- são compostas de múltiplos agentes (humanos, artificias ou ambos);
- são direcionadas a objetivos, que no entanto podem ser alterados, mal articulados e não compartilhados por todos os seus membros;
- podem afetar e ser afetadas pelo ambiente que as cerca;
- armazenam conhecimento, cultura, memória, história e capacidades distintas de qualquer agente individual;
- sua existência do ponto de vista legal é distinta daquela dos agentes individuais.

4.3 FINALIDADE DE ORGANIZAÇÕES DE AGENTES

Uma possível justificativa para a existência das organizações humanas, segundo Carley e Gasser (1999), é a de ultrapassar as limitações individuais dos agentes. Tais limitações podem ser classificadas em basicamente 4 (quatro) tipos distintos:

- 1. limitações cognitivas: agentes possuem uma racionalidade limitada (SI-MON, 1957), e portanto em certas situações precisam se agregar para alcançar melhor desempenho;
- 2. limitações físicas: devido à sua fisiologia, bem como à distribuição de recursos pelo ambiente, agentes devem necessariamente se coordenar com outros para realizar determinadas ações;
- 3. limitações temporais: agentes devem necessariamente se juntar a outros para atingir objetivos de longo prazo, cuja expectativa de cumprimento exceda a sua própria vida;
- 4. limitações institucionais: devido às limitações políticas ou legais de um agente individual, em certas ocasiões estes devem agir como um único agente corporativo para atingirem certos objetivos.

Como já discutido na seção 4.1, existem inúmeros modos através dos quais os agentes se organizam para ultrapassar estas limitações. Infelizmente, décadas de pesquisas na área apontam que não existe uma resposta correta para o problema

de projetar a melhor organização, dado um certo conjunto de agentes e objetivos a serem atingidos. Devido a este fato, foram desenvolvidas diversas teorias de contingências (GALBRAITH, 1973) para minimizar este problema.

Uma outra finalidade importante no projeto de organizações de agentes é a de bem conciliar a autonomia destes últimos, tal como introduzida na seção 2.1.4 com as restrições a eles impostas, conforme a definição apresentada na seção 4.1, de modo a dotá-los da capacidade de alterar seu comportamento em função de alterações dinâmicas no ambiente. Uma discussão mais detalhada sobre este aspecto, no âmbito do modelo \mathcal{M} OISE⁺, é apresentada na seção 6.2.1.

4.4 Taxonomia de organizações de agentes

Para melhor caracterizar uma organização de agentes segundo o contexto deste trabalho, a tabela 4.1 apresenta as principais dimensões que a caracterizam.

Eixo	Tipos de Organizações em Sistemas Multiagentes							
gênese		forr	$_{ m nais}$	emergentes				
foco	DEO	FUN		EST	DEO	FUN EST		EST
nível	CON	CON	ABS	ABS	CON	CON	ABS	ABS
comportamento	E D	E D	E D	E D	D	D	D	D

onde: DEO = deôntico, FUN = funcional, EST = estrutural, ABS = abstrato, CON = concreto, E = estático e D = dinâmico

Tabela 4.1: Taxonomia de organizações de agentes

Cada uma destas dimensões é brevemente discutida nas seções seguintes.

4.5 GÊNESE DAS ORGANIZAÇÕES DE AGENTES

Considere uma vez mais o cenário de interação de robôs, com as seguintes hipóteses sobre o pedido de auxílio de Clotaire a Berthold:

 quando Clotaire solicitou a Berthold que a ajudasse a carregar Millie de volta ao seu local de trabalho, ela utilizou o fato de que ele desempenha o papel de carregador de robôs e que está alocado para trabalhar nas oficinas 7 a 11, onde ela se encontrava; 2. Clotaire já desconfia há muito tempo que Berthold tem uma certa "queda" por ela, e solicitou a sua ajuda supondo que Berthold não recusaria um pedido seu.

Existe uma diferença fundamental, relativa à organização dos robôs, entre as duas hipóteses acima. Na primeira, existe *objetivamente* uma organização formal, descrita por exemplo no estatuto da empresa que controla a oficina, que prescreve funções, tarefas, direitos e deveres dos agentes que a constituem. No segundo caso, tal organização não existe.

Conte e Castelfranchi (1992) apresentam 2 (dois) modelos que caracterizam o processo de gênese das organizações¹, descritas a seguir.

4.5.1 Organizações formais

As organizações formais impõem aos agentes um problema prévio a resolver. As interações entre os agentes são impostas (modelo descendente ou "top-down"), e sua situação de interação, conforme discutida na seção 3.2, é pré-estabelecida como uma hipótese de partida e resulta numa colaboração simples ou coordenada. A organização também limita a utilização dos métodos de interação, descritos na seção 3.5, que "guiam" os agentes para atingir o objetivo global do sistema.

4.5.2 Organizações emergentes

Já nas chamadas organizações emergentes, os agentes não têm necessariamente um objetivo comum a atingir a priori. As interações entre os agentes emergem (modelo ascendente ou "bottom-up") dinamicamente como forma de atingirem seus próprios objetivos.

Outros autores, como Lemaître e Excelente (1998), denominam a metodologia de projeto de sistemas com as características acima respectivamente como centrado nas organizações e centrado nos agentes.

Como já discutido no capítulo 3, usualmente se utiliza o termo coalizão, ou coligação, para caracterizar as organizações emergentes. Tipicamente, tais orga-

¹Na verdade, neste trabalho os autores discutem a gênese das interações sociais.

nizações podem ser encontradas tanto em SMA reativos como em SMA cognitivos (EPSTEIN; AXTELL, 1996; RESNICK, 1997), enquanto as organizações formais dizem respeito exclusivamente aos SMA cognitivos.

Relevância: Neste trabalho, os capítulos 5 e 8 apresentam respectivamente a descrição e a implementação de uma técnica, denominada CBD, que permite a gênese de organizações emergentes. Já os capítulos 6 e 9 apresentam respectivamente modelos que permitem projetar organizações formais, bem como implementações que ilustram a sua utilização prática.

4.6 FOCO DAS ORGANIZAÇÕES DE AGENTES

Em (FOX et al., 1998), apresenta-se uma ontologia detalhada para organizações no contexto empresarial. Esta ontologia, representada na figura 4.1, contém os seguintes elementos principais:

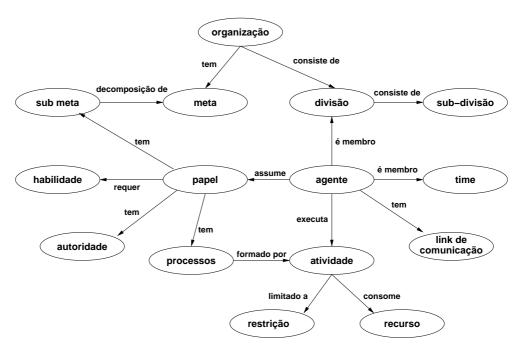


Figura 4.1: Ontologia organizacional (FOX et al., 1998).

 Organização, que consiste de várias divisões e subdivisões, um conjunto de agentes alocados nestas divisões, um conjunto de papéis que os agentes assumem e um conjunto de objetivos;

- Papéis, que representam protótipos de funções a serem desempenhadas pelos agentes na organização;
- Atividades, que consomem recursos, devem obedecer a restrições e são os meios para atingir os objetivos da organização;
- Agentes, que são membros de uma divisão da empresa, assumem um ou mais papéis, realizando as atividades associadas e podem se comunicar com outros agentes.

Praticamente todos os trabalhos da área utilizam, mesmo que com pequenas modificações, os conceitos apresentados na taxonomia acima. Enquanto alguns trabalhos privilegiam os aspectos estruturais das organizações (relativo aos papéis), outros se dedicam mais ao estudo dos seus aspectos funcionais (relativos às atividades). Um terceiro foco de interesse, usualmente mais relacionado com as organizações emergentes, são os chamados aspectos deônticos, relacionados com a noção de compromisso. Exemplos de trabalhos cujo foco principal resida em cada um destes aspectos são apresentados a seguir.

4.6.1 ASPECTOS ESTRUTURAIS

Na ontologia apresentada por Fox (1998, apud HÜBNER 2003), as seguintes propriedades estão associadas a cada papel:

- um conjunto de objetivos que o agente que assume o papel deve buscar;
- um conjunto de processos (tarefas) que definem como os objetivos podem ser alcançados;
- um conjunto de autoridades que o agente necessitará para alcançar os objetivos (direitos sobre determinados recursos, por exemplo);
- um conjunto de habilidades que o agente que assume o papel deve possuir;
- um conjunto de restrições na execução dos processos;
- um conjunto de recursos necessários para o papel ser desempenhado.

Sobre os papéis também existem as seguintes relações: (i) hierarquia, pois um papel pode ser subordinado a outro e (ii) especialização, já que um papel pode especializar outro papel e herdar seus direitos, obrigações, autoridades etc. No cenário de interação de robôs, o papel de soldador é subordinado ao papel de chefe da oficina, e o papel de microsoldador é uma especialização do papel de soldador.

Um modelo de organização cujo foco é o aspecto estrutural é o modelo Aaladin (FERBER; GUTKNECHT, 1998), representado na figura 4.2.

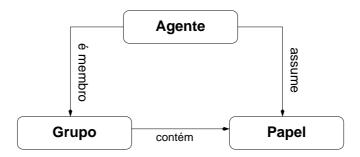


Figura 4.2: Modelo organizacional Aaladin (FERBER; GUTKNECHT, 1998).

Neste modelo, a organização é definida como um conjunto de grupos que possuem uma determinada estrutura. Cada grupo contém um conjunto de papéis necessários ao seu funcionamento e um conjunto de agentes membros. Os papéis são representações abstratas para as funções que os agentes disponibilizam. Nenhuma restrição é feita quanto à arquitetura interna dos agentes: um agente é considerado simplesmente como uma entidade ativa e comunicante que assume papéis nos grupos onde é membro. Um outro modelo interessante, cujo foco são os aspectos estruturais, é apresentando em (PATTISON; CORKILL; LESSER, 1987).

4.6.2 ASPECTOS FUNCIONAIS

A figura 4.3 (DECKER, 1996) mostra o modelo organizacional TAEMS, cuja ênfase recai nos aspectos estruturais de uma organização de agentes, e cuja noção principal é a de *tarefa*.

Segundo o modelo, existem 3 (três) visões distintas de uma tarefa: (i) visão objetiva, que representa a tarefa de modo completo; (ii) visão subjetiva, representada nos agentes que a executarão, onde somente é representada a parte da

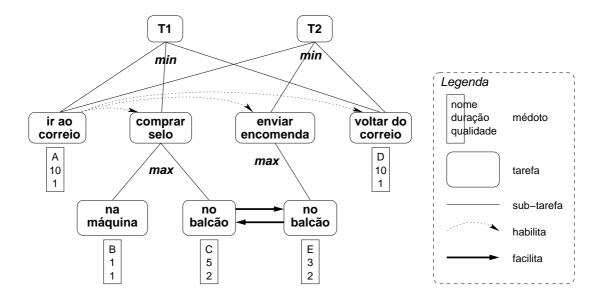


Figura 4.3: Modelo organizacional TAEMS (DECKER, 1996).

tarefa que lhes cabe realizar e (iii) visão generativa, que permite gerar as visões anteriores, dado um determinado problema a solucionar. A partir destas visões geradas, podem-se calcular informações estatísticas sobre a tarefa.

Tarefas são estruturadas ainda através de duas relações: sub-tarefa e efeitos colaterais.

A relação de sub-tarefa forma uma árvore hierárquica, cujas folhas representam métodos capazes de serem executados pelos agentes. Tais métodos podem possuir atributos como a qualidade desejada do resultado de sua execução e sua duração. A qualidade de uma tarefa pode ser calculada como a soma, o valor mínimo ou o valor máximo da qualidade de suas sub-tarefas. No exemplo representado na figura 4.3, a qualidade da tarefa T2 é calculada como o valor mínimo da qualidade de suas sub-tarefas enviar encomenda e voltar do correio.

A relação de efeitos colaterais ("nonlocal-effects") permite identificar se a execução de uma tarefa altera positivamente ou negativamente a qualidade de outra tarefa. No exemplo representado na figura 4.3, a execução de uma das sub-tarefas de comprar selo e enviar encomenda, realizadas no balcão da agência de correio, facilita a execução da outra. No caso em que se relacionam tarefas sendo realizadas por agentes diferentes, observa-se que tal qualidade depende do grau de coordenação entre as ações realizadas por estes agentes. Embora não seja o objetivo do modelo, a visão generativa permite construir diversas políticas de

escalonamento possíveis, para permitir avaliar quais estruturas de tarefa são as mais adequadas, robustas e rápidas para solucionar um determinado problema.

Outros modelos interessantes, cujo foco são os aspectos funcionais, são os modelos STEAM (TAMBE, 1997) e Shared Plans (GROSZ; KRAUS, 1996).

4.6.3 Aspectos deônticos

Segundo outros autores, uma outra noção que pode caracterizar uma ação coletiva e, conseqüentemente, formar uma organização de agentes é a noção de compromisso.

A arquitetura BDI, discutida na seção 2.1.6, se baseia na noção de intenção. Num trabalho sem dúvida alguma seminal para o domínio de SMA, Cohen e Levesque (1987) apresentam um modelo onde a perenização de uma escolha ("choice"), através de um compromisso ("commitment"), constitui uma intenção ("intention = choice + commitment"). O mecanismo pelo qual um agente decide como e quando deve abandonar suas intenções (e portanto os compromissos associados a elas) é chamado de estratégia de compromisso. Agentes podem utilizar estratégias distintas, em relação aos meios e aos fins a serem atingidos, tais como (RAO; GEORGEFF, 1991, apud WOOLDRIDGE 2002):

- compromisso cego ("blind commitment"), também chamado de fanático, quando o agente mantém a sua intenção até que acredite que ela foi atingida;
- compromisso persistente ("single-minded commitment"), quando o agente mantém a sua intenção até que acredite que ela foi atingida ou que, em caso contrário, enquanto acredite que ainda seja possível atingí-la;
- compromisso aberto ("open-minded commitment"), onde o agente mantém a sua intenção até que considere que ainda seja possível atingí-la.

A relação entre a noção de compromisso e de organização aparece com o trabalho posterior destes pesquisadores (LEVESQUE; COHEN; NUNES, 1990), onde definem a noção de compromisso conjunto ("joint commitment"). Resumidamente, este modelo considera que a relação de pertinência a um grupo numa ação coletiva requer do agente a adoção de uma responsabilidade em relação aos outros membros do grupo, em particular quanto à manutenção de suas intenções

em realizar determinadas ações. Os autores argumentam que esta atitude deve ser conhecida de todos os membros do grupo. Assim, quando um agente resolver não mais manter suas intenções, por exemplo por ter descoberto que não é mais possível atingí-las, ele deve advertir os outros membros do grupo deste fato.

Retornando mais uma vez ao cenário da interação dos robôs, Berthold, ao ajudar Clotaire a transportar Millie, tem representado em sua mente que (i) está realizando a tarefa de carregar a cabeça de Millie; (ii) esta tarefa é parte de uma ação coletiva que envolve Clotaire e (iii) tem um compromisso com esta última em relação a esta ação. Caso Berthold necessite momentaneamente abandonar sua intenção de carregar Millie, por exemplo para recarregar sua bateria, ele deve avisar Clotaire, que é membro do grupo responsável por esta ação coletiva.

Um outro modelo interessante, cujo foco são os aspectos deônticos, é apresentado em (DIGNUM; DIGNUM, 2001).

Relevância: Os modelos \mathcal{M} OISE e \mathcal{M} OISE, apresentados respectivamente nas seções 6.1 e 6.2, contemplam os aspectos funcionais, estruturais e deônticos de uma organização de agentes.

4.7 NÍVEIS DE ORGANIZAÇÕES DE AGENTES

Nos modelos que utilizam a noção de papel organizacional, coexistem sempre 2 (duas) organizações:

4.7.1 Organizações abstratas

Uma organização abstrata é estática e atemporal, e contém apenas agentes abstratos, ou seja, os papéis organizacionais. Seu objetivo é o estabelecimento de normas e restrições aos agentes que fazem parte da organização.

4.7.2 Organizações concretas

Já uma organização concreta é dinâmica e real, e contém agentes que desempenham os pápéis a eles associados. Pode ser entendida como uma "instanciação" de uma organização abstrata, associando agentes a papéis.

No modelo Aaladin (FERBER; GUTKNECHT, 1998), usa-se o termo estrutura organizacional para caracterizar a organização abstrata, formada por grupos e papéis. A organização concreta surge durante o funcionamento da sociedade, quando os agentes formam grupos e assumem determinados papéis.

Relevância: Os modelos \mathcal{M} OISE e \mathcal{M} OISE, apresentados no capítulo 5, denominam tais organizações abstratas e concretas respectivamente de especificação organizacional e entidade organizacional.

4.8 Representação das organizações de agentes

De modo semelhante aos argumentos apresentados na seção 3.3, relativos à representação da interação, agentes podem estar conscientes ou não das organizações nas quais estão inseridos. No caso de SMA reativos, estas organizações, caso existam, podem ter sido "introjetadas" nos agentes pelo projetista do sistema. No caso dos SMA cognitivos, obviamente a representação subjetiva, na mente dos agentes, destas relações organizacionais lhes dão mais recursos para se adaptarem a ambientes dinâmicos e imprevisíveis, como discutido na seção seguinte.

A figura 4.4 mostra os possíveis casos de presença/ausência de representação de aspectos organizacionais na mente de agentes cognitivos, nos casos de organizações formais e emergentes (HÜBNER, 2003). No caso (a), a organização é do tipo emergente e não está representada na mente dos agentes, sendo que apenas um observador externo pode modelar o comportamento do sistema como sendo organizado. Já no caso (b), a organização que emerge durante a interação dos agentes é representada internamente por estes últimos. Entretanto, os agentes podem construir representações organizacionais distintas, representadas na figura por quadrados de tons diferentes. No caso (c), o sistema possui uma organização formal, mas os agentes não têm capacidade para representá-la. Finalmente, no caso (d) tem-se que os agentes podem tanto representar a organização formal, quanto a organização emergente. Novamente, os agentes podem ter representações distintas devido ao seu processo de percepção.

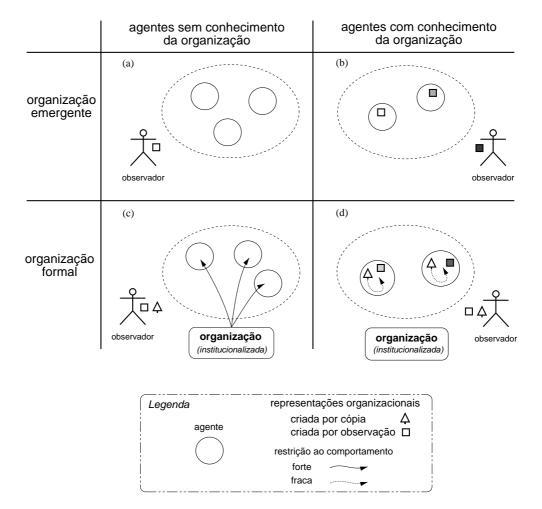


Figura 4.4: Representação intencional e não intencional de organizações formais e emergentes (HÜBNER, 2003).

Relevância: Nas aplicações dos modelos \mathcal{M} OISE e \mathcal{M} OISE⁺, apresentadas no capítulo 9, os agentes podem representar internamente os modelos das organizações nas quais estão inseridos.

4.9 Reorganização

As organizações formais² podem ter 2 (dois) comportamentos temporais distintos:

- organizações estáticas, que mantém sempre sua estrutura e funcionalidade inalteradas;
- organizações dinâmicas, que eventualmente alteram sua estrutura e funcionalidade caso seja necessário.

Se, por um lado, o estabelecimento de uma organização formal otimiza o comportamento de um SMA para que este atinja os seus objetivos, por outro lado a sua manutenção permanente diminui a flexibilidade do sistema, principalmente quando este deve se adaptar a mudanças no seu ambiente.

O processo de reorganização é definido aqui da seguinte maneira:

Um processo de reorganização é aquele que altera uma organização formal em seus aspectos estuturais, funcionais, ou deônticos, seja em seu nível abstrato, seja em seu nível concreto.

O processo de reorganização normalmente é muito complexo, e seus efeitos podem ser inúmeros. Tomando como exemplo mais uma vez o domínio da interação dos robôs, um possível processo de reorganização, destinado a melhor monitorar o desempenho de Berthold como carregador, poderia ser o seguinte:

• alteração estrutural, nível abstrato: a criação de um novo papel de coordenador de transporte, e de um novo grupo de transporte, contendo dois transportadores e um coordenador de transporte;

²Esta discussão não se aplica às organizações emergentes, já que as coalizões se desfazem automaticamente quando terminam sua ação coletiva.

- alteração funcional, nível abstrato: a tarefa de transporte de robôs deve ser realizada pelo novo grupo de transporte, sendo que a decisão sobre o uso ou não da maca deve ser tomada pelo coordenador, mas a velocidade de transporte e o itinerário pelos robôs transportadores;
- alteração estrutural, nível concreto: a adoção deste novo papel de coordenador pelo robô Yun, e a criação do respectivo grupo juntamente com Clotaire e Berthold.
- alteração funcional, nível concreto: Berthold deve solicitar agora a Yun a
 permissão para a utilização da maca de transporte, como também decidir
 a velocidade e o itinerário a ser adotado.

No que diz respeito ao processo de reorganização em si, ele pode ser realizado de 3 (três) modos distintos (HÜBNER, 2003):

- pré-definido, quando o processo de reorganização já está previsto no próprio modelo organizacional, por exemplo através de instantes precisos para alteração de suas funcionalidades;
- controlado, quando apesar de não se conhecer previamente quando a reorganização irá acontecer, sabe-se de antemão como o processo de reorganização irá ocorrer;
- emergente, quando n\u00e3o h\u00e1 qualquer controle expl\u00edcito por parte do sistema sobre o processo de reorganiza\u00e7\u00e3o, que pode ocorrer, por exemplo, por iniciativa individual de um agente.

No caso de uma reorganização controlada, o processo pode ser dividido em 4 (quatro) etapas (PATTISON; CORKILL; LESSER, 1987): (i) monitoração; (ii) projeto; (iii) seleção e (iv) implementação. Cada etapa, por sua vez, pode ser executada de modo endógeno, quando um ou mais agentes do próprio sistema se encarregam de realizar o processo³, ou de modo exógeno, quando o usário do SMA realiza as alterações necessárias.

³Este processo também é chamado de auto-organização.

 $4.10 \quad Conclus\~oes$ 73

Relevância: O modelo \mathcal{M} OISE⁺, apresentado na seção 6.2, foi concebido explicitamente para poder servir de base a um processo de reorganização, cujas características principais são detalhadas na seção 6.2.6.

4.10 Conclusões

Um primeiro resultado importante, observado neste capítulo, e complementar ao apresentado na seção 3.7 é o seguinte:

As organizações de agentes limitam as suas interações, garantindo que estes atinjam seus objetivos globais de modo otimizado.

Um segundo resultado, relativo às discussões apresentadas nos capítulos que compõem esta parte I, pode ser enunciado como se segue:

Agentes cognitivos autônomos, imersos num SMA aberto, podem aumentar sua eficiência e adaptabilidade a alterações no ambiente, caso tenham a capacidade de representar explicitamente e explorar, através de mecanismos de raciocínio adequados, as capacidades de outros agentes e as eventuais organizações em que estejam envolvidos.

Este segundo resultado será ilustrado através dos modelos e aplicações que são apresentados nas partes II e III que se seguem.

Parte II

Modelos

5 Modelos de Raciocínio Social

Neste capítulo, são apresentados respectivamente nas seções 5.1 e 5.2 os modelos sociais \mathcal{S}_{RM} e \mathcal{S}_{RM}^+ . Tais modelos são descritos de forma resumida, de modo a realçar suas características principais. A descrição formal dos mesmos encontrase respectivamente nos anexos A e B. Na seção 5.3, estes modelos são situados em relação à classificação proposta na seção 2.3.

Mais detalhes sobre os modelos \mathcal{S}_{RM} e \mathcal{S}_{RM} podem ser encontrados respectivamente em (SICHMAN et al., 1994; CONTE; SICHMAN, 1995; SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995a; SICHMAN; DEMAZEAU, 1995b; SICHMAN; DEMAZEAU, 1996; SICHMAN, 1997b; SICHMAN, 1998; SICHMAN; DEMAZEAU, 2001) e (CONTE; SICHMAN, 2002; SICHMAN; CONTE, 2002).

5.1 Modelo $S_{\rm RM}$

O modelo Srm (SICHMAN et al., 1994; CONTE; SICHMAN, 1995; SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995a; SICHMAN; DEMAZEAU, 1995b; SICHMAN; DEMAZEAU, 1996; SICHMAN, 1997b; SICHMAN, 1998; SICHMAN; DEMAZEAU, 2001) é baseado na chamada Teoria da Dependência, proposta em (CASTELFRANCHI, 1990; CASTELFRANCHI; MICELLI; CESTA, 1992). Esta teoria propõe respostas para duas questões fundamentais, presentes numa sociedade de agentes autônomos:

- 1. Problema da sociabilidade: por que um agente autônomo participa de interações sociais?
 - 2. Problema da adoção: como um agente consegue tornar o seu problema

 $5.1 \; Modelo \; S_{\rm RM}$

social, isto é, fazer com que seja adotado por outros agentes?

De acordo com a Teoria da Dependência, a resposta à primeira pergunta é a $dependência\ social$, enquanto que a resposta à segunda pergunta é o $poder\ social$. Basicamente, um agente ag_i depende de um agente ag_j caso exista entre eles uma interferência social, tal como descrita na seção 3.1, sendo que nesta situação o agente ag_j adquire um certo poder sobre o agente ag_i , já que sua ação pode facilitar ou impedir que o agente ag_i atinja seus objetivos. Caso os agentes tenham uma representação desta interferência social, a interação entre os agentes torna-se intencional, como discutido na seção 3.3.1 e como resultado podem surgir dinamicamente coalizões entre os agentes, estabelecendo assim organizações emergentes a partir de suas interações, como discutido nas seções 3.4 e 4.5.2.

Para compreender este fenômeno, é preciso introduzir os conceitos de ação social, delegação e adoção de objetivos, o que é feito na seqüência.

5.1.1 AÇÃO SOCIAL

Supondo que dois agentes ag_i e ag_j co-existam num mesmo ambiente e que g_m seja um objetivo de ag_i , podem-se distinguir as seguintes possíveis ações sociais de ag_i e de ag_j :

- exploração: se ag_j tiver também o objetivo g_m e estiver realizando uma ação para atingí-lo, basta a ag_i esperar que ag_j realize sua ação;
- influencia: caso a competência de ag_i não seja suficiente para atingir g_m , e que exista uma interferência social positiva entre ag_i e ag_j , ag_i pode tentar convencer ag_j a adotar este objetivo, ou seja, fazer com que g_m se torne também um objetivo de ag_i ;
- $agress\~ao$: caso ag_j possua um outro objetivo g_n incompatível com g_m , ag_i pode tentar impedir ag_j de atingí-lo;
- adoção: se, por alguma razão, ag_j deseja que ag_i atinja seu objetivo g_m , ele pode adotar este objetivo, ou seja, g_m torna-se um objetivo de ag_j pela simples razão de este ser originalmente um objetivo de ag_i .

 $5.1 \; Modelo \; S_{\rm RM}$

5.1.2 Delegação e adoção de objetivos

Uma consequência fundamental da interferência social é a alteração dos objetivos atingíveis por um determinado agente. Por exemplo, na presença de ag_j , ag_i pode atingir um determinado objetivo que seria inatingível se estivesse sozinho, pois ag_j pode interferir positivamente por meio da realização de uma determinada ação.

Neste caso, diz-se que ag_i não é $autônomo^1$ para atingir um determinado objetivo, sendo dependente de ag_i .

Resumidamente, a Teoria da Dependência diz que ag_i deve raciocinar sobre o melhor modo de delegar a ação de que necessita a ag_j : garantir que esta será realizada por ag_j e que este último esteja consciente de que a realiza para que ag_i alcance seu objetivo. Assim, ag_i pode utilizar um mecanismo de planejamento social para tentar realizar uma ação social de influência: convencer ag_j a adotar o objetivo de realizar a determinada ação para que ag_i alcance seu objetivo. Uma vez aceita a adoção, ag_j firma um compromisso junto a ag_i para realizar a ação desejada, como discutido na seção 4.6.3.

Uma possibilidade é oferecer a ag_j a realização de alguma ação em troca, seja para atingir o mesmo objetivo ou para atingir outro objetivo de ag_j . No primeiro caso, diz-se que ocorre uma cooperação entre ag_i e ag_j , e no segundo caso diz-se que ocorre um $escambo\ social$ entre estes agentes. Estas duas noções são especializações da situação de interação denominada colaboração simples, apresentada na seção 3.2.2.

5.1.3 MECANISMO DE RACIOCÍNIO SOCIAL

O modelo \mathcal{S}_{RM} propõe um mecanismo de raciocínio social, parte integrante da arquitetura de um agente cognitivo, que pode auxiliá-lo nos seguintes aspectos:

- 1. melhorar a sua *adaptação* a mudanças no ambiente, no contexto de SMA abertos (SICHMAN, 1995; SICHMAN, 1997b);
- 2. avaliar a susceptibilidade de outros agentes adotarem seus objetivos, e consequentemente melhorar o desempenho do processo de formação de coalizões

 $^{^1\}mathrm{Adota}\text{-se}$ aqui o terceiro significado do termo autonomia, apresentado na seção 2.1.4.

 $5.1 \; Modelo \; S_{\rm RM}$

(SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995b);

3. auxiliar na detecção e *revisão de crenças* a respeito dos outros agentes, de modo a manter uma consistência desta informação (SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995a; SICHMAN; DEMAZEAU, 1996).

Tais aspectos são melhor detalhados na seção 8.2.1, onde se apresenta o sistema $\mathcal{D}_{\text{EPINT}}$.

Apresentam-se a seguir, nas seções 5.1.4 a 5.1.11, os principais aspectos do modelo \mathcal{S}_{RM} .

5.1.4 Descrição externa

Para poder raciocinar socialmente, um agente deve representar internamente algumas propriedades dos outros agentes. Tais propriedades são armazenadas numa estrutura denominada descrição externa, que é composta por várias entradas, cada uma delas contendo as seguintes informações sobre um agente em particular: (i) os objetivos que este agente deseja alcançar; (ii) as ações que ele sabe desempenhar; (iii) os recursos que ele controla e (iv) os planos que ele pode utilizar para atingir seus objetivos². Os planos são compostos de uma seqüência de ações instanciadas, que consistem de uma ação com uma lista associada de recursos para desempenhá-la (que pode ser vazia). Objetivos, ações e recursos têm também uma medida quantitativa associada, referente respectivamente à sua importância e custo. Um exemplo de descrição externa simplificada de uma sociedade de agentes, contendo apenas seus objetivos, planos e ações, é mostrada na tabela 5.1, adaptada de (SICHMAN; CONTE, 2002). Esta sociedade é composta por 8 (oito) agentes, e nas seções seguintes é utilizada como ilustração do mecanismo de raciocínio social.

Estas informações sobre os outros agentes podem ser adquiridas através de 3 (três) fontes distintas, conforme as funcionalidades de um agente discutidas na seção 2.1.1: percepção do ambiente, recepção de uma mensagem ou inferência. Esta fonte da informação também é armazenada na descrição externa, e serve

²Considera-se no modelo que os agentes têm armazenados planos pré-definidos para atingir seus objetivos.

Descrição Externa							
Agente	Objetivos	Planos	Ações				
ag_1	g_1	$p_{111} = a_1, a_2, a_4$ $p_{112} = a_1, a_5$	$a_1,\ a_9$				
		$p_{112} = a_1, a_5$ $p_{13} = a_3, a_9$					
	g_5	$p_{15}=a_1,a_{10}$					
	g_8	$p_{18}=a_1, a_7$					
	g_9	$p_{19}=a_9,a_{11}$					
ag_2	g_2	$p_{22}=a_2, a_6$	a_2				
ag_3	g_3	$p_{33}=a_3, a_8$	a_2, a_3				
ag_4	g_4	$p_{44} = a_4, a_7$	a_4				
ag_5	g_1	$p_{51}=a_1,a_5$	a_5, a_{10}				
ag_6	g_6	$p_{66}=a_1, a_6$	a_6, a_7				
ag_7	g_7	$p_{77} = a_7, a_9$	a_7				
ag_8	g_{10}	$p_{810} = a_8$	a_8, a_{10}				

Tabela 5.1: Exemplo de descrição externa no modelo S_{RM} .

para detectar eventuais inconsistências que os agentes podem ter uns em relação aos outros. Tal aspecto não será aqui abordado³, e é descrito em mais detalhes em (SICHMAN, 1995; SICHMAN; DEMAZEAU, 1996).

5.1.5 Relações de dependência

Utilizando a informação presente na descrição externa, um agente pode calcular suas relações de dependência em relação aos outros agentes.

Um agente ag_i é dito autonômo para o objetivo g_m se e somente se: (i) ag_i deseja atingir o objetivo g_m ; (ii) existe um plano p_s cuja execução atinja g_m tal que todas as suas ações podem ser desempenhadas por ag_i^4 . No exemplo apresentado na tabela 5.1, o agente ag_8 é autônomo para o objetivo g_{10} , já que segundo o plano p_{810} para atingir este objetivo só é necessária a ação a_8 , que o agente ag_8 sabe executar.

Caso não seja autônomo para um dado objetivo, um agente ag_i é dito de-pendente para este objetivo. O fato de ser dependente, porém, não significa que

 $^{^3{\}rm A}$ descrição formal deste aspecto, entretanto, pode ser encontrada no anexo A.

⁴O modelo formal é um pouco mais complexo, pois leva em conta a fonte dos planos, bem como a existência de recursos e ações necessários à execução de um plano. Simplificadamente, aqui se restringe o raciocínio às ações, sendo que o agente utiliza seu próprio repertório de planos.

 $5.1 \; Modelo \; \mathcal{S}_{\mathrm{RM}}$

exista necessariamente um agente que possa executar a ação de que ag_i necessita. Esta situação é representada por uma relação de dependência. Assim, um agente ag_i é dito dependente de um agente ag_j para o objetivo g_m se e somente se: (i) ag_i tem o objetivo g_m ; (ii) ag_i é dependente para o objetivo g_m (iii) existe um plano p_s cuja execução atinja g_m e no qual ag_j pode realizar alguma ação de que ag_i necessita. No exemplo apresentado na tabela 5.1, o agente ag_1 é dependente do agente ag_5 para o objetivo g_1 , já que segundo o plano p_{112} para atingir este objetivo é necessária a ação a_5 , que ag_1 não sabe realizar, mas que ag_5 sabe executar.

5.1.6 Redes de dependência

5.1.7 Relações de OU-dependência e de E-dependência

Relações de dependência podem ser ainda aditivas ou alternativas (CONTE; VENEZIANO; CASTELFRANCHI, 1998). Nas redes de dependência, observa-se que existem casos nos quais uma determinada ação pode ser realizada por mais de um agente. No caso do exemplo representado na figura 5.1, este é o caso da ação a_2 , que pode ser realizada pelos agentes ag_2 e ag_3 . Este tipo de dependência é denominado OU-dependência.

Por outro lado, em certas situações os agentes necessitam de várias ações para atingirem seus objetivos, que podem não ser necessariamente realizadas por um único agente. Novamente examinando a figura 5.1, este é o caso do objetivo g_1 , já que segundo o plano p_{111} o agente ag_1 necessita das ações a_2 e a_4 . Este tipo de dependência é denominado E-dependência.

A noção de OU-dependência normalmente diminui o grau de dependência de um agente, já que ele tem alternativas para tentar obter o auxílio que necessita,

5.1 Modelo $S_{\rm RM}$

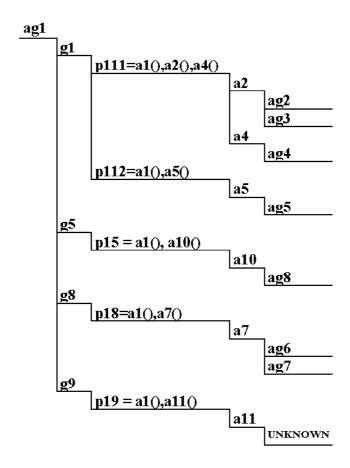


Figura 5.1: Exemplo de rede de dependência no modelo S_{RM} (SICHMAN; CONTE, 2002).

 $5.1 \; Modelo \; \mathcal{S}_{\mathrm{RM}}$

e portanto aumenta o seu poder de barganha. De modo oposto, a noção de Edependência aumenta o grau de dependência de um agente, caso se considere que este é função do número de ações envolvidas.

5.1.8 Relações de dependência mútua e recíproca

Sempre que um agente ag_i deduzir que é dependente de ag_j para algum de seus objetivos, é interessante que ele possa avaliar se esta relação de dependência é unilateral ou bilateral. Utilizando as definições de (CASTELFRANCHI; MICEL-LI; CESTA, 1992), denomina-se dependência mútua (MD) a situação onde dois agentes ag_i e ag_j dependem um do outro para o mesmo objetivo g_m . No exemplo apresentado na tabela 5.1, existe uma dependência mútua entre os agentes ag_1 e ag_5 para o objetivo g_1 , já que segundo o plano p_{112} para atingir este objetivo são necessárias as ações a_1 e a_5 , sendo que cada um sabe realizar uma destas ações. Trata-se de um exemplo da existência de complementaridade entre os agentes, como discutido na seção 2.1.5.

Agentes também podem ser complementares de outro modo. Tomando-se novamente o exemplo apresentado na tabela 5.1, o agente ag_1 depende do agente ag_3 para o objetivo g_1 , já que segundo o plano p_{111} para atingir este objetivo é necessária a ação a_2 , que ag_1 não sabe realizar mas que ag_3 sabe executar. Por outro lado, existe também outra relação de dependência entre estes dois agentes: o agente ag_3 também depende do agente ag_1 , mas para o objetivo g_3 , já que segundo o plano p_{13} para atingir este objetivo é necessária a ação a_9 , que ag_3 não sabe realizar mas que ag_1 sabe executar.

Esta situação é denominada no modelo de dependência recíproca (RD), quando dois agentes ag_i e ag_j dependem um do outro, porém, para objetivos distintos g_m e g_n .

As situações de dependência mútua e recíproca, se detectadas pelos agentes, podem levá-los respectivamente a estabelecer situações de interação do tipo cooperação e escambo social, conforme discutido na seção 5.1.2.

5.1.9 SITUAÇÕES DE OBJETIVO

Chama-se situação de objetivo uma relação entre um agente ag_i e um objetivo g_m , que pode assumir 4 (quatro) valores distintos:

- 1. No Goal (NG): o agente ag_i não deseja atingir o objetivo g_m ;
- 2. No Plans (NP): o agente ag_i tem o objetivo g_m , mas não sabe como atingí-lo;
- 3. Autonomous (AUT): o agente ag_i tem o objetivo g_m e pelo menos um plano no qual pode desempenhar todas as ações;
- 4. **Dependent** (**DEP**): o agente ag_i tem o objetivo g_m , mas em todos os planos ele necessita do auxílio de outros agentes para realizar determinada ação.

No exemplo apresentado na tabela 5.1, a situação de objetivo de ag_1 para o objetivo g_1 é DEP. Nesta situação, um agente pode calcular qual a sua situação de dependência em relação aos outros agentes, como mostrado a seguir.

5.1.10 SITUAÇÕES DE DEPENDÊNCIA

Quando um agente ag_i raciocina socialmente, ele normalmente utiliza seu próprio repertório de planos para inferir suas relações de dependência. Entretanto, como os agentes também armazenam em suas descrições externas os planos dos outros agentes, em certos casos pode ser útil a utilização por ag_i dos planos de ag_j , por exemplo, para avaliar se uma possível relação de dependência entre ag_i e ag_j , unilateral ou bilateral, também é de conhecimento de ag_j .

Um agente infere *localmente* uma relação de dependência bilateral se ele utiliza apenas seus próprios planos para inferir esta relação. Por outro lado, um agente infere *mutualmente* uma relação de dependência bilateral se ele utiliza, além de seus próprios planos, também os planos do agente do qual depende para inferir esta relação⁵.

⁵Diferentemente da semântica usual de crença mútua em IA, aqui o termo se refere à situação onde o agente que raciocina acredita que o agente do qual depende também está consciente desta relação.

Considere-se então dois agentes ag_i e ag_j , onde o agente que está raciocinando socialmente é ag_i . Se ag_i inferir DEP como sua situação de objetivo para g_m , existem 6 (seis) diferentes situações de dependência entre ag_i e ag_j que podem ocorrer:

- 1. Independence (IND): utilizando seus próprios planos, ag_i infere que não depende de ag_i para g_m ;
- 2. Locally Believed Mutual Dependence (LBMD): utilizando seus próprios planos, ag_i infere uma dependência mútua entre ele e ag_j para g_m , mas não pode chegar à mesma conclusão utilizando os planos de ag_i ;
- 3. Mutually Believed Mutual Dependence (MBMD): utilizando seus próprios planos, ag_i infere uma dependência mútua entre ele e ag_j para g_m , e chega à mesma conclusão utilizando os planos de ag_i ;
- 4. Locally Believed Reciprocal Dependence (LBRD): utilizando seus próprios planos, ag_i infere uma dependência recíproca entre ele e ag_j para g_m e g_n , mas não pode chegar à mesma conclusão utilizando os planos de ag_j ;
- 5. Mutually Believed Reciprocal Dependence (MBRD): utilizando seus próprios planos, ag_i infere uma dependência recíproca entre ele e ag_j para g_m e g_n , e chega à mesma conclusão utilizando os planos de ag_j ;
- 6. Unilateral Dependence (UD): utilizando seus próprios planos, ag_i infere que depende de ag_j para g_m , mas este não depende de ag_i para nenhum de seus objetivos.

A título de ilustração, o mecanismo de raciocínio social do agente ag_1 , no exemplo apresentado na tabela 5.1, infere que (i) existe uma UD entre ele e o agente ag_8 para o objetivo g_5 , pois ele necessita da ação a_{10} mas nada pode oferecer em troca a este agente; (ii) existe uma MBMD entre ele e o agente ag_5 para o objetivo g_1 , já que cada um deles é capaz de desempenhar uma das ações, a_1 e a_5 , necessárias para atingir este objetivo, e ambos estão conscientes deste fato, que pode ser inferido a partir de p_{112} e de p_{51} e (iii) existe uma LBRD

⁶No modelo, os agentes podem ter planos distintos para atingirem o mesmo objetivo. Uma vez estabelecida uma coalizão, pode haver uma negociação para definir o plano a ser utilizado, como discutido na seção 3.5.5.

entre ele e o agente ag_3 para os objetivos g_1 e g_3 , já que ele necessita da ação a_2 para atingir g_1 e pode oferecer a ag_3 em troca, segundo o seu plano p_{13} , a ação a_9 , mas ag_3 não sabe que esta ação pode ser útil para atingir este objetivo, já que ela está ausente de seu plano p_{33} .

5.1.11 COALIZÕES BASEADAS EM DEPENDÊNCIA

O mecanismo de raciocínio social acima descrito pode ser utilizado como meio de implementar o modelo de resolução cooperativa de problemas, descrito na seção 3.4, especialmente no que se refere às fases de detecção do potencial de cooperação e formação de coalizão. Tal técnica, denominada de Coalizões Baseadas em Dependência (CBD), é composta dos seguintes passos:

- 1. o agente constrói uma rede de dependência, onde se encontram representadas conjuntamente todas as suas relações de dependência;
- 2. após a construção desta rede, o agente escolhe um objetivo g_m a atingir e um plano p_s para este fim;
- 3. o agente então calcula a sua situação de objetivo para g_m ;
- 4. caso a sua situação de objetivo para g_m seja AUT, ele executa as ações necessárias para atingir seu objetivo;
- 5. caso a sua situação de objetivo para g_m seja DEP, ele calcula a sua situação de dependência para este objetivo em relação a todos os outros agentes dos quais depende e escolhe um possível parceiro que possa ajudá-lo para este fim, enviando-lhe uma proposta de coalizão, onde solicita a execução da ação a_p , necessária segundo o plano p_s para atingir o objetivo g_m , e oferece realizar em troca⁷ a ação a_q , necessária para que o parceiro possa atingir o objetivo g_n^8 ;
- 6. caso o parceiro aceite a proposta, a coalizão é formada, e as ações são realizadas;
- 7. caso o parceiro escolhido não aceite a proposta, retorna-se ao passo (5) para escolher outro parceiro.

 $^{^{7}}$ Com exceção do caso em que exista uma dependência unilateral (UD).

⁸Nos caso de dependência mútua (MBMD, LBMD), têm-se $g_m = g_n$.

8. caso não haja mais parceiros disponíveis no passo (5), retorna-se ao passo (2) para escolher um outro plano para atingir o mesmo objetivo;

- 9. caso não haja mais planos disponíveis para o objetivo corrente no passo (2), escolhe-se um outro objetivo para atingir;
- 10. caso não haja mais objetivos a atingir, o processo termina.

Considera-se aqui exclusivamente o caso onde haja apenas OU-dependências entre os agentes. Uma extensão para o caso de E-dependências é apresentado a seguir, na seção 5.1.12.

No modelo apresentado acima, um plano é dito factivel caso todas as ações necessárias à sua execução estiverem disponíveis na sociedade de agentes. Como o modelo se destina a ser utilizado num SMA aberto, conforme discutido na seção 2.1.4, não se pode garantir que um determinado plano seja factível num determinado momento. No exemplo apresentado na tabela 5.1, caso o agente ag_4 saia do sistema, o plano p_{111} deixa de ser factível, pois nenhum outro agente é capaz de realizar a ação a_4 . Durante o passo (2) acima, dado um determinado objetivo corrente, um agente escolhe executar o plano de menor custo dentre os planos factíveis.

Um objetivo é dito atingível caso exista pelo menos um plano factível para atingí-lo. No exemplo apresentado na tabela 5.1, o objetivo g_9 não é atingível, pois nenhum agente á capaz de realizar a ação a_{11} . Durante o passo (2) acima, o agente escolhe o objetivo de maior importância dentre os objetivos atingíveis.

Em relação à escolha de parceiros, estabeleceu-se um critério de escolha de parceiros baseado na combinação de 2 (dois) sub-critérios:

- situações de dependência inferidas mutualmente são preferidas àquelas inferidas localmente, pois intuitivamente não haveria o esforço adicional de convencimento do parceiro;
- 2. situações de dependência mútua são preferíveis àquelas de dependência recíproca, pois um agente não benevolente não teria interesse em não cumprir o compromisso (problema de reciprocidade).

Segundo este critério, as situações de dependência obedeceriam à relação de ordem parcial, $MBMD > \{MBRD, LBMD\} > LBRD > UD$, onde x > y indica que x é preferível a y.

Assim, tomando mais uma vez o exemplo apresentado na tabela 5.1, supondo que o agente ag_1 tenha escolhido atingir o objetivo g_5 , através da execução do plano p_{15} , ele iria preferir formar uma coalizão com o agente ag_5 do que com o agente ag_8 , já que conforme o critério de escolha de parceiros acima descrito tem-se que LBRD > UD.

Por outro lado, um agente ag_i só aceita uma proposta de coalizão de um agente ag_j , onde lhe é oferecida uma ação a_q para alcançar um determinado objetivo g_n , nos casos em que (i) g_n for realmente um objetivo de ag_i ; (ii) ag_i realmente não saiba realizar a ação a_q ; (iii) a situação de objetivo de ag_i para g_n for NP ou então (iv) a situação de objetivo de ag_i para g_n for DEP e ag_j pertence ao grupo de agentes que seriam seus parceiros preferenciais, caso ag_i tivesse de propor uma coalizão para atingí-lo.

O sistema \mathcal{D} EPINT, descrito na seção 8.2, implementa esta utilização do mecanismo de raciocínio social, através da utilização de um protocolo de comunicação desenvolvido para este fim, que se encontra representado na figura 8.3.

5.1.12 Coalizões baseadas em dependência extendidas

A técnica de CBD acima apresentada foi posteriormente extendida (DAVID, 1998; DAVID; SICHMAN; COELHO, 1999; DAVID; SICHMAN; COELHO, 2000) para poder tratar casos onde haja *E-dependência*, ou seja, onde as coalizões envolvam múltiplos parceiros, considerando que os agentes podem necessitar de múltiplas ações para atingir um determinado objetivo.

Inicialmente, cabe notar que as situações de dependência apresentadas na seção 5.1.10 sempre utilizam como ponto de partida os planos do próprio agente. Em certas ocasiões, entretanto, pode ser estrategicamente interessante iniciar o raciocínio a partir dos planos do outro agente. Assim, acrescentou-se a possibilidade de um agente inferir remotamente uma relação de dependência, unilateral ou bilateral, caso ele utilize exclusivamente os planos do agente do qual supostamente depende para inferir esta relação.

Adicionalmente, o mecanismo de escolha de parceiros foi generalizado, permitindo que agentes ofereçam um conjunto de objetivos e ações, ao invés de um único objetivo e uma única ação. Tal fato se justifica pela complexidade inerente ao processo de coordenação de ações, como discutido na seção 3.5.4. Por exemplo, se um agente ag_i necessitar de duas ações a_p e a_q para atingir o objetivo g_m , e se puder obter o auxílio de ag_j e ag_k para a_p e de ag_j e ag_l para a_q , o fato de escolher ag_j minimiza o possível custo de coordenação, já que este agente pode realizar as duas ações de que ag_i necessita. No exemplo apresentado na tabela 5.1, caso o agente ag_2 pudesse desempenhar também a ação a_4 , o agente ag_1 poderia preferir formar uma coalizão unicamente com ele para atingir o objetivo g_1 , ao invés de necessitar também envolver o agente ag_4 .

Esta extensão altera também os passos do processo de formação de coalizão apresentados na seção 5.1.11, pois caso estejam sendo formadas coalizões com mais de dois agentes, quando uma ação não puder ser obtida (por exemplo, se os possíveis parceiros não aceitarem a proposta), os parceiros que já foram contactados para realizar outras ações devem ser avisados de que a coalizão não irá mais se formar.

O sistema \mathcal{D} epint⁺, descrito na seção 8.3, implementa esta utilização do mecanismo de raciocínio social, através da utilização de um novo protocolo de comunicação desenvolvido para levar em conta esta extensão, e que se encontra representado na figura 8.3.

5.1.13 Coalizões baseadas em dependência e rede contratual

Conforme discutido na seção 3.5.3, as técnicas de RC e de CBD correspondem a métodos não informados e informados para divisão de tarefas em SMA. Apresenta-se em (ITO, 1999; ITO; SICHMAN, 2000) um estudo comparativo destas duas técnicas quanto ao fluxo global de mensagens do SMA, cujos resultados são mostrados na figura 5.2.

Supõe-se que no caso das CBD, os agentes inicialmente passam por uma fase de apresentação, quando informam uns aos outros seus objetivos, planos, ações e recursos. Já na RC, esta fase não ocorre. Por outro lado, no caso da RC o agente que necessita formar uma coalizão sempre envia mensagens a todos os

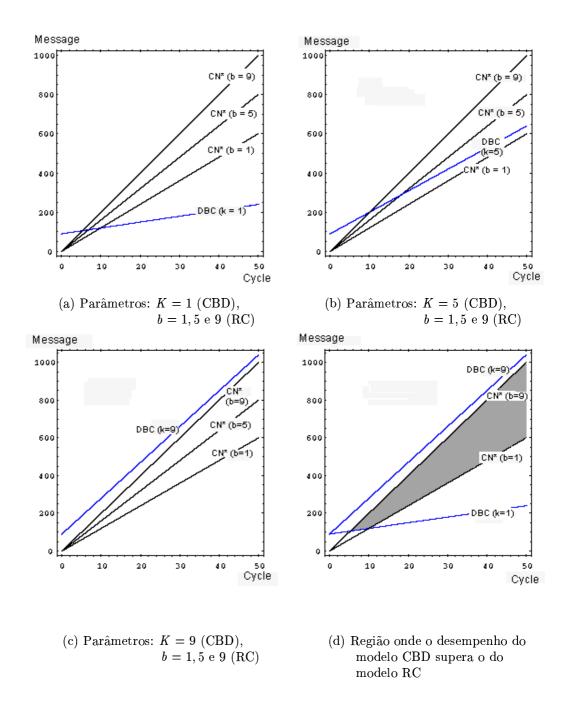


Figura 5.2: Comparação do fluxo global de mensagens entre as técnicas CBD e RC, para um sistema com 10 agentes (ITO; SICHMAN, 2000).

agentes do sistema, enquanto que nas CBD uma única mensagem é enviada por vez, destinada ao possível parceiro escolhido.

Nos gráficos apresentados na figura 5.2, é simulada uma sociedade composta por 10 agentes. O fator k representa nas CBD o número de possíveis parceiros a quem propostas de coalizão foram enviadas. Por exemplo, se o primeiro parceiro escolhido aceitou a proposta de coalizão (bom raciocínio social), o valor de k vale 1. Esta situação é mostrada no gráfico (a). Num outro extremo, mostrado no gráfico (c), o valor de K é 9, o que significa que o agente teve que enviar propostas a todos os outros agentes, sendo que apenas o último deles aceitou a proposta (raciocínio social ruim). Analogamente, o fator b representa na RC o número de ofertas recebidas pelo contratante. Observa-se nos gráficos (a) e (b), onde neste segundo se adota um valor médio de k igual a 5, que existe sempre um determinado ciclo de interação crítico, a partir do qual o fluxo de comunicação global nas CBD é menor do que o da RC. A comparação entre o desempenho das técnicas para os valores de b e k iguais a 1, 5 e 9 é sumarizada no gráfico (d).

Os resultados mostram que caso o raciocínio social seja efetivo (valor próximo de 1), existirá sempre um determinado ciclo crítico a partir do qual fluxo de comunicação é menor nas CBD do que na RC. Deste modo, se demonstra que a técnica de CBD é mais adequada em contextos onde os mesmos agentes interagem repetidamente, em diversos ciclos de formação de coalizões.

5.2 Modelo SRM⁺

O modelo SRM se destina a analisar possíveis interações entre dois agentes, cuja interferência social seja positiva. Em certas ocasiões, entretanto, existem possíveis padrões de interação social que envolvem mais de dois agentes, e onde as relações de dependência são circulares. O modelo SRM⁺ (CONTE; SICHMAN, 2002; SI-CHMAN; CONTE, 2002) foi concebido para tratar estes casos, como mostrado a seguir.

5.2.1 Relações de dependência circulares

Tomando mais uma vez o exemplo apresentado na tabela 5.1, seja a situação na qual o agente ag_1 escolha o objetivo g_1 e o plano p_{111} para atingir este objetivo. Neste caso, existe uma relação de E-dependência, discutida na seção 5.1.7, já que ag_1 necessita das ações a_2 e a_4 . Supondo ainda que ele tente formar uma coalizão com os agentes ag_2 e ag_4 , para que realizem respectivamente as ações a_2 e a_4 . Se calcular sua situação de dependência em relação a estes dois agentes para o objetivo g_1 , ag_1 obterá UD para os dois casos. Trata-se de uma situação não muito confortável para ag_1 , pois seus parceiros não teriam porque ajudá-lo a atingir g_1 , caso não fossem benevolentes.

Entretanto, analisando melhor as relações de dependência entre os agentes do exemplo, observa-se que ag_2 necessita da ação a_6 para atingir g_2 , enquanto que ag_4 necessita da ação a_7 para atingir g_4 . Os agentes ag_6 e ag_7 são capazes de realizar respectivamente estas ações, e ambos dependem de ag_1 para atingir seus objetivos. O primeiro necessita da ação a_1 para atingir g_6 e o segundo necessita da ação a_9 para atingir g_7 .

Caso consiga raciocinar estrategicamente deste modo, ag_1 pode propor uma coalizão a estes quatro agentes, sendo que todos irão atingir seus objetivos. Este padrão de dependência é denominado AMONG-dependência e é detalhado na seção 5.2.4. Antes disto, entretanto, é necessário extender o modelo de redes de dependência para poder lidar com estas situações. O modelo de grafos de dependência foi criado com esta finalidade.

5.2.2 Grafos de dependência

Um grafo de dependência ("dependence graph") é um grafo orientado com as seguintes propriedades:

- 1. o conjunto de vértices é formado pela união de 4 (quatro) subconjuntos disjuntos de vértices, que representam agentes, objetivos, planos e ações;
- 2. o conjunto de arestas ligam os vértices nas seguintes condições:
 - (a) o vértice que representa o agente ag_i é ligado ao vértice que representa o objetivo g_j somente se o agente tenha este objetivo;

(b) o vértice que representa o objetivo g_i é ligado ao vértice que representa o plano p_i somente se a execução deste plano atingir este objetivo;

- (c) o vértice que representa o plano p_i é ligado ao vértice que representa a ação a_j somente se esta ação for necessária a este plano e o agente que se encontra na raiz do caminho a que pertence esta aresta não souber realizar esta ação;
- (d) o vértice que representa a ação a_i é ligado ao vértice que representa o agente ag_i somente se este for capaz de realizar esta ação.

A figura 5.3 ilustra o grafo de dependência associado ao exemplo apresentado na tabela 5.1, adotando algumas simplificações⁹.

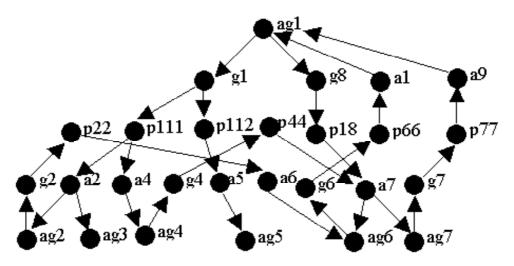


Figura 5.3: Exemplo de grafo de dependência no modelo S_{RM}^+ (SICHMAN; CONTE, 2002).

5.2.3 Grafos de dependência reduzidos

Como se pode perceber pela figura 5.3, grafos de dependência podem se tornar muito complexos, dificultando a visualização e o tratamento dos fenômenos de interação que se propõem a analisar e representar.

Assim, num primeiro momento, criou-se uma subclasse destes grafos, denominada grafos de dependência reduzidos. Nestes grafos reduzidos, cada agente tem

⁹Neste grafo, não foi representado o agente ag_8 , e tampouco os objetivos e planos de ag_3 e de ag_5 . Limitou-se também a representar apenas os objetivos g_1 e g_8 de ag_1 .

apenas um único objetivo a atingir e um único plano para atingí-lo. Deste modo, os vértices que representam objetivos e planos desaparecem e o grafo resultante fica mais simplificado. O objetivo para o qual um agente necessita de uma ação é agora colocado como um rótulo da respectiva aresta. Um exemplo de grafo de dependência reduzido é apresentado na figura 5.4, grafo este gerado a partir do grafo completo mostrado na figura 5.3.

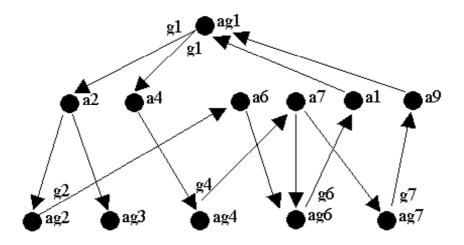


Figura 5.4: Exemplo de grafo de dependência reduzido no modelo \mathcal{S}_{RM}^+ (SICHMAN; CONTE, 2002).

Os fenômenos de interação social que podem ser representados por este grafo reduzido são mais limitados do que aqueles que podem ser representados no grafo completo, mas tais fenômenos são bastante interessantes, como mostrado a seguir.

5.2.4 AMONG-DEPENDÊNCIA

Dado um grafo reduzido de dependência, define-se uma AMONG-dependência como a situação onde existe um ciclo no grafo envolvendo mais de dois agentes, como mostra a figura 5.5. Neste grafo, a AMONG-dependence é caracterizada pelo ciclo que envolve os agentes ag_1 , ag_2 e ag_7 . Neste grafo, existe também uma dependência recíproca, como discutida na seção 5.1.8, envolvendo os agentes ag_3 e ag_4 .

Caso exista numa sociedade de agentes uma relação de OU-dependência ou de E-dependência envolvendo agentes que também se encontram numa relação de AMONG-dependência, alguns fenômenos interessantes podem ser caracterizados,

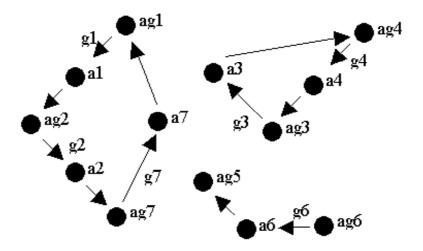


Figura 5.5: Exemplo de AMONG-dependência (SICHMAN; CONTE, 2002).

descritos nas seções a seguir.

5.2.4.1 AMONG-DEPENDÊNCIA E OU-DEPENDÊNCIA

Caso um agente ag_j tenha uma relação de AMONG-dependência com o conjunto de agentes Ag_i e esteja também numa relação de OU-dependência com um subconjunto de Ag_i , existirão pelo menos 2 (dois) ciclos distintos no grafo de dependência, originados no vértice ag_j , que poderão gerar obstáculos para a formação de coalizões estáveis.

Desigualdade em AMONG-dependência e OU-dependência é mostrado na figura 5.6. Neste tipo de situação, sempre existirá um agente mais útil (solicitado) do que dependente, como o agente ag_6 no caso da figura. O fato do agente ag_1 poder optar entre formar uma coalizão com ou sem a presença do agente ag_2 introduz uma desigualdade no poder de decisão da rede, em benefício do agente OU-dependente. No entanto, uma vez estabelecida a coalizão, o agente ag_6 irá realizar apenas uma ação (a_1 ou a_2). Este tipo de desigualdade ameaça a formação de coalizões estáveis.

Incompatibilidade em AMONG-dependência Um segundo exemplo de combinação entre AMONG-dependência e OU-dependência é mostrado na figu-

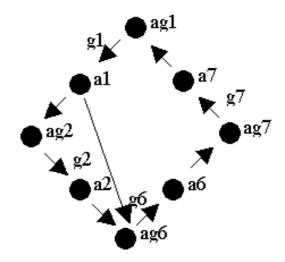


Figura 5.6: Desigualdade em AMONG-dependência (SICHMAN; CONTE, 2002).

ra 5.7. Neste caso particular, o agente OU-dependente é util ao mesmo número de agentes dos quais depende, como o agente ag_1 no caso da figura. Tem-se então duas coalizões incompatíveis, uma envolvendo ag_1 , ag_2 e ag_6 , e outra envolvendo ag_1 , ag_4 e ag_7 . Mais uma vez, a presença de uma OU-dependência prejudica a formação de coalizões estáveis.

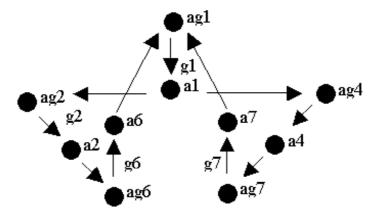


Figura 5.7: Incompatibilidade em AMONG-dependência (SICHMAN; CONTE, 2002).

5.2.4.2 AMONG-DEPENDÊNCIA E E-DEPENDÊNCIA

Caso um agente ag_j tenha uma relação de AMONG-dependência com o conjunto de agentes Ag_i e esteja também numa relação de E-dependência com um subcon-

junto de Ag_i , também existirão pelo menos 2 (dois) ciclos distintos no grafo de dependência, mas que diferentemente do caso anterior não são necessariamente nocivos à formação de coalizões estáveis.

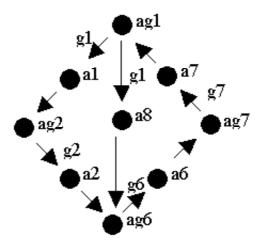


Figura 5.8: AMONG-dependência frágil (SICHMAN; CONTE, 2002).

AMONG-DEPENDÊNCIA FRÁGIL Um primeiro exemplo de combinação entre AMONG-dependência e E-dependência é mostrado na figura 5.8. Novamente, sempre existirá um agente mais útil (solicitado) do que dependente, como o agente ag_6 no caso da figura. Diferentemente do caso da desigualdade em OU-dependência, nesta situação a coalizão somente se estabelece caso ag_6 aceite executar duas ações (a_2 e a_8), em troca de apenas uma (a_6). Obviamente, tal situação prejudica a estabilidade das coalizões.

AMONG-DEPENDÊNCIA FORTE Já o segundo exemplo de combinação entre AMONG-dependência e E-dependência, mostrado na figura 5.9, é completamente distinto. Nesta situação, a E-dependência fortalece a formação de coalizões, pois gera dois ciclos no grafo, envolvendo ag_1 , ag_2 e ag_6 , e outro envolvendo ag_1 , ag_4 e ag_7 , que não são somente compatíveis, como também complementares.

5.2.5 GROUP-DEPENDÊNCIA

Uma *GROUP-dependência* é um caso particular de uma AMONG-dependência quando é válida a seguinte cláusula, denominada cláusula de equidade ("equity clause"):

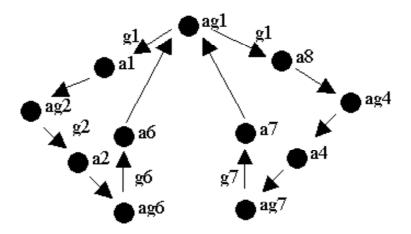


Figura 5.9: AMONG-dependência forte (SICHMAN; CONTE, 2002).

Cláusula de Eqüidade: Para todo agente ag_i que se encontra em AMONGdependência com o grupo de agentes Ag_j , o subconjunto de Ag_j dos quais ag_i depende tem a mesma cardinalidade do subconjunto de Ag_j que contém os agentes que dependem de ag_i .

Em outras palavras, cada agente é igualmente útil e dependente, como o caso mostrado na figura 5.9.

GROUP-DEPENDÊNCIA DECENTRALIZADA Suponha-se que a seguinte cláusula, denominada cláusula egalitária ("egalitarian clause") seja válida, adicionalmente à cláusula de equidade:

Cláusula Egalitária: Para todo agente ag_i que se encontra em AMONGdependência com o grupo de agentes Ag_j , a cardinalidade dos subconjuntos de Ag_j dos quais ag_i depende e que dependem de Ag_j é a mesma.

Em outras palavras, todos os agentes têm a mesma importância no grupo. Este não é ocaso da situação mostrada na figura 5.9, pois o agente ag_2 é util e depende de 2 (dois) agentes, enquanto todos os outros são úteis e dependem de um único agente. O subconjunto formado pelos agentes ag_1 , ag_2 e ag_7 , apresentado na figura 5.5, é um exemplo desta situação.

AMONG-DEPENDÊNCIA CENTRALIZADA Na situação mostrada na figura 5.9, o agente ag_2 tem um papel mais central que os outros agentes na coalizão. O caso limite desta situação ocorre quando todos os agentes envolvidos na GROUP-dependência dependem diretamente de um agente central, como é o caso do agente ag_1 na situação mostrada na figura 5.10.

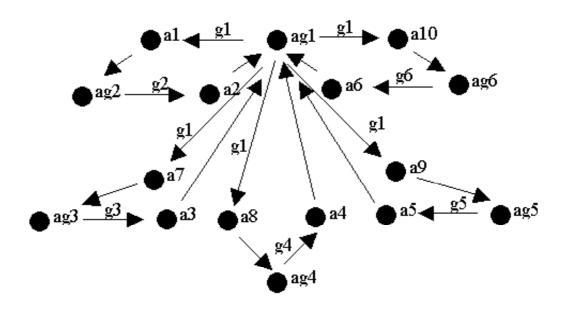


Figura 5.10: GROUP-dependência centralizada (SICHMAN; CONTE, 2002).

5.2.6 COLLECTIVE-DEPENDÊNCIA

Uma COLLECTIVE-dependência representa para uma GROUP-dependência o mesmo que uma dependência mútua representa para uma dependência recíproca no caso de dois agentes, como discutido na seção 5.1.8. Uma COLLECTIVE-dependência ocorre num conjunto de agentes AMONG-dependentes quando cada agente depende dos outros para atingirem um mesmo objetivo, como representado na figura 5.11.

Ao contrário de haver uma precisa distinção entre as noções de GROUPdependência e COLLECTIVE-dependência, tais noções encontram-se num eixo contínuo de tipos de relações de interação entre vários agentes. Tais tipos variam de acordo com as características dos grafos de dependência associados, tais como 5.3 Conclusões 99

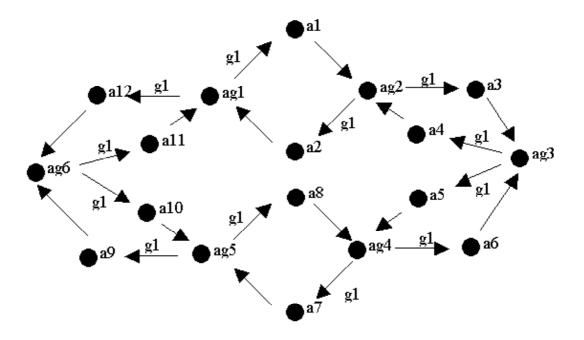


Figura 5.11: COLLECTIVE-dependência (SICHMAN; CONTE, 2002).

o número de arestas, o grau de centralização/decentralização e a existência ou não de um objetivo comum a todos os agentes envolvidos.

5.3 Conclusões

A figura 5.12 mostra os modelos de raciocínio social \mathcal{S}_{RM} e \mathcal{S}_{RM}^+ classificados segundo os eixos básicos propostos na seção 2.3.

Em relação à sua finalidade, ambos os modelos podem ser utilizados tanto numa perspectiva de resolução social, como numa perspectiva de simulação social. Esta afirmação pode ser comprovada através das aplicações \mathcal{D} EPINT e \mathcal{D} EPINT⁺ (resolução social, modelo \mathcal{S} RM) e \mathcal{D} EPNET (simulação social, modelo \mathcal{S} RM), descritas no capítulo 8. Os dois modelos se situam no nível de descrição teórico. Finalmente, ambos os modelos abordam a dimensão de interação entre agentes da metodologia Vogais.

Considerando os aspectos referentes às características de agentes e SMA, abordados no capítulo 2, ambos os modelos pressupõem que os agentes que os utilizam sejam autônomos, cognitivos e sociais, conforme a taxonomia apresentada na seção 2.1.2, e cuja arquitetura seja deliberativa ou híbrida, conforme discutido

 $5.3 \quad Conclus\~oes$ 100

na seção 2.1.6. Segundo ainda a taxonomia de SMA proposta na seção 2.2.2, os modelos podem ser aplicados em SMA abertos ou não, de baixa granularidade (poucos agentes), compostos por agentes homogêneos ou heterogêneos. Obviamente, neste segundo caso os agentes devem ter associados a si o mecanismo de raciocínio social proposto pelos modelos.

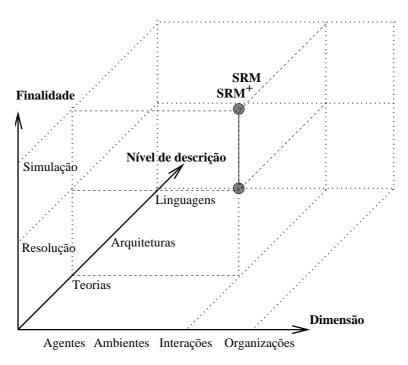


Figura 5.12: Classificação dos modelos de raciocínio social.

No que diz respeito à discussão sobre a interação entre agentes, apresentada no capítulo 3, os modelos foram propostos para explorar e estruturar situações de interação cuja interferência social seja positiva, tais como a colaboração simples ou coordenada, discutidas na seção 3.2.2. Em particular, os modelos podem ser utilizados como meio de implementar o modelo de resolução cooperativa de problemas, descrito na seção 3.4, especialmente no que se refere às fases de detecção do potencial de cooperação e formação de coalizão. A técnica de CBD, descrita nas seções 5.1.11 e 5.1.12, ilustra precisamente este ponto. No que se refere aos métodos de interação descritos na seção 3.5, estão relacionados com a divisão de tarefas. Implicitamente, as propostas de formação de coalizão utilizam um protocolo de comunicação entre agentes, tal como discutido na seção 3.6.4.

Considerando ainda as propriedades de organizações de agentes, discutidas no capítulo 4, ambos os modelos têm como foco a dimensão funcional de uma

5.3 Conclusões 101

organização emergente, conforme a taxonomia apresentada na seção 4.4.

Finalmente, no que diz respeito à representação das interações e da organização, discutidas nas seções 3.3 e 4.8, ambos os modelos podem ser utilizados como parte da arquitetura de agentes cuja interação seja intencional ou não intencional.

6 Modelos de Raciocínio Organizacional

Neste capítulo, são apresentados respectivamente nas seções 6.1 e 6.2 os modelos organizacionais \mathcal{M} OISE e \mathcal{M} OISE⁺. Tais modelos são descritos de forma resumida, de modo a realçar suas características principais. A descrição formal dos mesmos encontra-se respectivamente nos anexos C e D. Na seção 6.3, estes modelos são situados em relação à classificação proposta na seção 2.3.

Mais detalhes sobre os modelos \mathcal{M} OISE e \mathcal{M} OISE+ podem ser encontrados respectivamente em (HANNOUN et al., 1998; HANNOUN et al., 1999; HANNOUN et al., 2000; HANNOUN, 2002) e (HÜBNER, 2003; HÜBNER; SICHMAN; BOISSIER, 2002a; HÜBNER; SICHMAN; BOISSIER, 2002b; HÜBNER; SICHMAN; BOISSIER, 2002c).

6.1 Modelo \mathcal{M} oise

Segundo o modelo Moise (Hannoun et al., 1998; Hannoun et al., 1999; Hannoun et al., 2000; Hannoun, 2002), o objetivo de uma organização, no contexto de SMA, é o de restringir a autonomia de um agente. Tais restrições são expressas no modelo através de permissões, obrigações e proibições para as diversas atividades que os agentes podem desempenhar no sistema.

O modelo Moise se articula em torno de 3 (três) níveis:

- 1. *nível individual*, que define as restrições sobre as tarefas e as responsabilidades dos agentes;
- 2. *nível social*, que define as restrições sobre as conexões e a comunicação global entre os agentes;

3. *nível coletivo*, que define as restrições sobre os agrupamentos e subconjuntos de agentes.

Em cada um destes níveis, propõem-se diversos *conceitos* do modelo que estruturam e organizam as atividades dos agentes.

Ortogonalmente a estes níveis, no modelo Moise são diferenciadas as organizações abstratas e concretas, tal como discutido na seção 4.7:

- 1. estrutura organizacional, composta pelo conjunto das restrições formais que especificam a organização do sistema;
- 2. entidade organizacional, composta pelo SMA resultante da aplicação destas restrições formais.

Nas seções seguintes, os conceitos definidos em cada um destes níveis são melhor detalhados. Em particular, será utilizado como exemplo ilustrativo destes conceitos um cenário simplificado de um programa de pós-graduação, que será também adotado na aplicação \mathcal{R} AISORG, descrita na seção 9.1.

6.1.1 Definições preliminares

Semelhantemente aos modelos sociais apresentados no capítulo 5, o modelo MOISE se baseia nos mesmos conceitos de base presentes na descrição externa dos agentes, apresentada na seção 5.1.4. Agentes visam atingir objetivos, e para isto recorrem a planos que especificam a seqüência de ações a serem executadas, sendo que tais ações podem eventualmente necessitar de recursos associados.

Para poder tratar mais facilmente conjuntos de elementos com características similares, o modelo introduz ainda a noção de *classe*:

Definição 6.1 (Classe) Uma classe define um conjunto de elementos de determinado tipo (objetivos, planos, ações, recursos, protocolos de comunicação e atos de fala) com características similares.

Já a comunicação entre agentes se dá através da troca de *atos de fala*, estruturados em *protocolos* de comunicação, conforme discutido nas seções 3.6.1 e 3.6.4.

6.1.2 NÍVEL INDIVIDUAL

No nível individual do modelo \mathcal{M} OISE, são definidas as restrições dos agentes em relação às tarefas que podem ou devem executar, e às ações e aos recursos que lhes são permitidos utilizar no contexto destas tarefas. Este nível é caracterizado pelos conceitos de papel e de missão, descritos a seguir.

6.1.2.1 Papel

No cenário do programa de pós-graduação, os cargos de professor, secretária, bem como o "status" de aluno têm associados a si um conjunto de deveres, obrigações e direitos que definem pap'eis a serem desempenhados pelos agentes pertencentes a esta organização.

Definição 6.2 (Papel) Um papel especifica o comportamento esperado de um agente na organização, bem como o conjunto de atividades que ele tem o direito de exercer.

Por exemplo, um agente no papel de « Professor » deve satisfazer (noção de obrigação) o objetivo de « Ensinar », e adicionalmente tem a possibilidade (noção de permissão) de satisfazer o objetivo « OrientarAlunos ». Alguns planos, ações e recursos lhe são permitidos, e outros lhe são proibidos. Por exemplo, ele pode sempre utilizar o recurso « Giz » nas suas diferentes atividades.

O conceito de papel não permite capturar todas as sutilezas referentes às restrições impostas a um agente por uma organização. No cenário do programa de pós-graduação, o professor pode, por exemplo, utilizar o recurso « Sala-101 » para suas aulas, mas não pode utilizar para este fim o recurso « Sala-52 », reservadas às reuniões do Conselho. Assim, é necessário contextualizar as restrições organizacionais, através de um novo conceito.

Assim, um papel no modelo \mathcal{M} OISE é definido como um conjunto de missões que lhe são associadas, sendo que cada uma destas missões tem uma natureza (obrigação ou permissão).

6.1.2.2 MISSÃO

Uma $miss\tilde{a}o$ é definida no modelo $\mathcal{M}OISE$ do seguinte modo:

Definição 6.3 (Missão) Uma missão restringe uma atividade associada a um papel em termos de permissões, proibições e obrigações sobre os objetivos, planos, ações e recursos que podem ser utilizados e que estejam associados a esta atividade.

A tabela 6.1, inspirada em (HANNOUN, 2002), apresenta um exemplo de 2 (duas) missões, onde intervêm os papéis de « Secretária » e de « Candidato » ao programa de Doutorado. A missão « m₁ », relativa ao papel do « Candidato », somente lhe autoriza a execução do plano « p_1 », destinado a atingir o objetivo $\ll g_1 \gg (\ll Inscrição Doutorado \gg)$. Tal plano define a execução da ação $\ll a_1 \gg$ (« Preparar Documentação Inscrição »), que necessita de quatro recursos distintos, seguido da execução da ação $\ll a_2 \gg (\ll Registrar Inscrição \gg)$ e da ativação do sub-objetivo « g_2 » (« TratarDocumentaçãoInscrição »). A execução da ação $\ll a_2 \gg$ e do objetivo $\ll g_2 \gg$ lhe é proibida, pois nem este objetivo nem esta ação pertencem ao conjunto de objetivos e ações permitidos desta missão. Assim, tanto este objetivo como esta ação devem ser realizados no contexto de outra missão $\ll m_2 \gg$, associada ao papel de $\ll Secret\'aria \gg$, que permite a execução de ambos. Este exemplo mostra como uma tarefa global pode ser distribuída entre diferentes papéis da organização. Note-se também que a natureza da missão « m_2 » é obrigatória (O) para o papel de « Secretária », enquanto que a natureza da missão $\ll m_1 \gg \acute{\rm e}$ facultativa (P) para o papel do $\ll Candidato \gg$, no sentido de que este pode desejar ou não se inscrever num programa de Doutorado.

Uma relação importante definida no modelo é a relação de sub-missão. Uma missão m_j é uma sub-missão de outra missão m_i se m_j realiza um sub-plano de m_i , isto é, se dentro do contexto de m_j pode-se executar uma ação, um plano ou mesmo atingir um objetivo também definidos em m_i . No exemplo da tabela 6.1, a missão m_2 é sub-missão de m_1 , pois para realizar m_1 é necessário atingir o objetivo g_2 , que pertence à missão m_2 . Neste exemplo em particular, não somente m_2 é uma sub-missão de m_1 (por causa de g_2), como também m_1 depende de m_2 , pois g_2 não é permitido no contexto de m_1 . Esta relação é utilizada no cálculo de dependências organizacionais, descrito na seção 6.1.6.

Papel	$Candidato = \{(\mathbf{P}, m_1)\}$		
Missão	$m_1 = \langle \{g_1\}, \{p_1\}, \{a_1\}, \{Any_r\} \rangle$		
Objetivos			
	$g_2:\ Tratar Documenta arxii ao Inscrição$		
Planos	$p_1(g_1) = a_1(r_1, r_2, r_3, r_4); \ a_2; \ [g_2]$		
Ações	$a_1: Preparar Documenta ção Inscrição$		
Recursos	$oxed{r_1: \'Ultimo Diploma \qquad r_2: Hist\'orico Gradua artile{ao}}$		
	$r_3: Foto 3x4 \hspace{1cm} r_4: C\'{o}pia Identidade$		

Papel	$Secret\'{a}ria = \{(\mathbf{O}, m_2)\}$		
Missão	$m_2 = \langle \{g_2\}, \{p_2\}, \{a_2, a_3, a_4\}, \{Any_r\} \rangle$		
Objetivos	$g_2:\ Tratar Documenta arxiia o Inscrição$		
	$g_3: \ Estudar Documenta c$ $ ilde{ao} Inscriccio$		
Planos	$p_2(g_2) = [g_3]; (a_3 \mid a_4)$		
Ações	$a_2: Registrar Inscrição$		
	$oxed{a_3: RecusarInscriç\~ao} oxed{a_4: AceitarInscriç\~ao}$		

Tabela 6.1: Exemplos de papéis e missões no modelo Moise.

6.1.3 NÍVEL SOCIAL

Uma vez fixadas as restrições, no nível individual, sobre as tarefas, ações e recursos que os agentes podem utilizar em suas atividades, torna-se necessário estruturar as possíveis interações¹ que podem ocorrer entre os diversos papéis de uma organização. Tais restrições se dão através de relações ("links").

Definição 6.4 (Relação) Uma relação organizacional qualifica a interação esperada entre dois papeís, sendo aplicável somente a algumas missões associadas aos papeís que dela participam.

O modelo Moise define 3 (três) tipos distintos de relações, descritos a seguir.

6.1.3.1 RELAÇÃO DE COMUNICAÇÃO

Uma relação de *comunicação* permite que dois papéis possam trocar mensagens, no contexto de uma missão específica. Tal relação também estabelece o protocolo de comunicação que deve ser seguido caso os dois papéis venham a interagir.

No exemplo do programa de pós-graduação, para que o procedimento de

¹Este termo é utilizado aqui num sentido mais restrito do que o adotado no capítulo 3.

inscrição possa ocorrer de modo adequado, deve haver uma relação de comunicação entre o papel de « Candidato », no contexto da missão « m_1 », e o papel de « Secretária », no contexto da missão « m_2 ».

6.1.3.2 Relação de autoridade

Uma relação de *autoridade* permite subordinar um papel a outro, dentro do contexto de uma missão específica. Tal relação é exercida através dos protocolos de comunicação que podem ser utilizados no contexto de uma missão, em particular através dos papéis que os agentes podem desempenhar nestes protocolos.

No exemplo do programa de pós-graduação, o papel de « Candidato » pode utilizar um protocolo onde ele emita uma solicitação de delegação ao papel de « Secretária », para que este último execute a ação « TratarDocumentaçãoInscrição ». A existência de uma relação de autoridade permite ao primeiro papel esta solicitação de delegação. Tal permissão, entretanto, não tem nenhum efeito relativo ao resultado da execução da ação.

6.1.3.3 Relação de conhecimento

Uma relação de conhecimento define o subconjunto da organização que um agente tem o direito de representar e eventualmente explorar em seu mecanismo de raciocínio. Em certos casos, tal relação se justifica por razões de segurança ou de confidencialidade, quando um agente não deve necessariamente conhecer a estrutura completa de uma organização.

6.1.4 NÍVEL COLETIVO

Os níveis individual e social não são ainda suficientes para caracterizar toda a riqueza das relações presentes em organizações complexas. No cenário do programa de pós-graduação, normalmente o agente que desempenha o papel de $\ll Professor \gg$ não tem uma relação de autoridade com todos os alunos do programa, mas somente com aqueles matriculados em seu curso e com seus orientados. Esta noção organizacional é capturada através do conceito de grupo:

Definição 6.5 (Grupo) Um grupo é uma outra forma de contextualizar as atividades

no seio de uma corporação. Ele define o contexto (equipe, classe, ...) no qual as missões e ligações de uma organização são válidas.

6.1.5 ESPECIFICAÇÃO E ENTIDADE ORGANIZACIONAL

Uma vez definidas as restrições nos níveis individual, social e coletivo, pode-se finalmente definir a organização abstrata (estrutura organizacional), bem como a organização concreta (entidade organizacional) que a implementa.

6.1.5.1 ESTRUTURA ORGANIZACIONAL

Uma estrutura organizacional estabelece todas as restrições descritas nas seções anteriores, de modo a restringir a autonomia dos agentes pertencentes à organização:

Definição 6.6 (Estrutura Organizacional) Uma estrutura organizacional define, num nível abstrato, as restrições organizacionais nos níveis individual, social e coletivo. Ela se constitui por um conjunto de papéis, de ligações e de grupos da organização.

6.1.5.2 Entidade organizacional

Uma entidade organizacional é uma instância concreta de uma estrutura organizacional:

Definição 6.7 (Entidade Organizacional) Uma entidade organizacional define, num nível concreto, as restrições organizacionais nos níveis individual, social e coletivo. Ela se constitui por um conjunto de duplas papel-agente e pelas instâncias de grupos dos quais os agentes participam.

A figura 6.1 representa uma estrutura e uma entidade organizacional no cenário do programa de pós-graduação. A estrutura organizacional é composta pelos papéis « Coordenador », « Professor », e « Aluno ». Sem entrar no mérito das missões associadas a estes papéis, existe uma relação de comunicação entre o papel de « Coordenador » e o papel de « Professor », e outra entre o papel de « Professor » e o papel de « Aluno ». Um grupo « turma » é composto por dois papéis, « Professor » e « Aluno ». A relação de comunicação entre estes

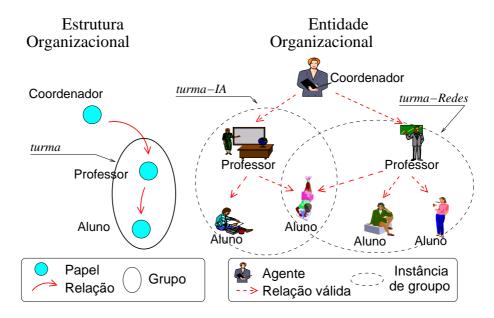


Figura 6.1: Exemplo de estrutura e entidade organizacional no modelo MOISE (HANNOUN, 2002).

dois papéis se situa no contexto deste grupo. Note-se ainda que um papel pode pertencer a vários grupos. Por exemplo, na entidade organizacional associada a esta estrutura, um mesmo aluno pertence a duas instâncias de grupo distintas, $\ll turma-IA \gg e \ll turma-Redes \gg$.

6.1.6 Dependência organizacional

Uma possível aplicação do modelo \mathcal{M} OISE é a sua utilização num contexto de $avaliação\ organizacional$, onde se pode detetar índices de desempenho e de coerência associados a certas características do modelo.

No trabalho apresentado em (HANNOUN et al., 1998; HANNOUN, 2002), é proposta uma noção de coerência organizacional, descrita na seção 6.1.7. Um primeiro passo para construir tal noção é a definição das chamadas relações de dependência organizacional. Tais relações, inspiradas no modelo SRM descrito na seção 5.1, podem ser associadas tanto ao nível da estrutura como ao nível da entidade organizacional, como mostrado a seguir.

6.1.6.1 Dependência na estrutura organizacional

No nível da estrutura organizacional, definem-se relações de dependência entre pap'eis, no escopo de missões, de modo semelhante àquelas definidas no modelo \mathcal{S}_{RM} , discutido na seção 5.1.5. Tais relações podem ser de 3 (três) tipos distintos: dependência de ações, de planos e de objetivos.

Dependência de ações na estrutura organizacional Um papel « r_i » depende de outro papel « r_j » em termos de ações se uma missão « m_q » de « r_j » for uma sub-missão de « m_p » de « r_i » e se esta última deve delegar uma ação instanciada « $a(r_1, \ldots, r_n)$ » que pode ser executada pela missão « m_q ».

No exemplo apresentado na tabela 6.1, o papel de « Candidato » depende do papel de « Secretária » em termos de ações, pois a missão « m_1 » associada ao primeiro papel deve delegar a realização da ação « a_2 » (« RegistrarInscrição ») para a missão « m_2 », associada ao segundo papel.

Dependência de planos na estrutura organizacional — Analogamente, um papel « r_i » depende de outro papel « r_j » em termos de planos se uma missão « m_q » de « r_j » for uma sub-missão de « m_p » de « r_i » e se esta última deve delegar um plano que pode ser ativado pela missão « m_q ».

Supondo que no exemplo apresentado na tabela 6.1 as missões dos papéis de « Candidato » e de « Secretária » fossem alteradas respectivamente para « $m_1 = \langle \{g_1\}, \varnothing, \{a_1\}, \{Any_r\} \rangle$ » e « $m_2 = \langle \{g_2\}, \{p_1, p_2\}, \{a_2, a_3, a_4\}, \{Any_r\} \rangle$ », o papel de « Candidato » dependeria então do papel de « Secretária » em termos de planos, pois a missão « m_1 » associada ao primeiro papel deveria delegar a realização do plano « p_1 » para a missão « m_2 », associada ao segundo papel.

DEPENDÊNCIA DE OBJETIVOS NA ESTRUTURA ORGANIZACIONAL Finalmente, um papel « r_i » depende de outro papel « r_j » em termos de objetivos se uma missão « m_q » de « r_j » for uma sub-missão de « m_p » de « r_i » e se esta última deve delegar um objetivo que pode ser ativado pela missão « m_q ».

No exemplo apresentado na tabela 6.1, o papel de « Candidato » depende do papel de « Secret'aria » em termos de objetivo, pois a missão « m_1 » associada ao

primeiro papel deve delegar a realização do objetivo « g_2 » (« TratarDocumentação Inscrição ») para a missão « m_2 », associada ao segundo papel.

6.1.6.2 Dependência na entidade organizacional

No nível da entidade organizacional, o modelo \mathcal{M} OISE se limita a definir a dependência de agentes em termos de ações.

Assim, um agente ag_x , que desempenha o papel « r_i », depende em termos de ações de outro agente ag_y no contexto de sua missão « m_p » se ag_y sabe desempenhar alguma ação instanciada « $a(r_1, \ldots, r_n)$ », necessária na missão « m_p », que ag_x não sabe realizar.

Novamente considerando o exemplo apresentado na tabela 6.1, suponha-se que os agentes ag_1 e ag_2 desempenhem respectivamente os papéis de « Candidato » e de « Secretária ». Neste caso, pode-se dizer que ag_1 depende de ag_2 , enquanto desempenha o papel de « Candidato », pois necessita que este último realize a ação ação « a_2 » (« RegistrarInscrição »).

6.1.7 Coerência organizacional

As relações de dependência organizacional, descritas na seção 6.1.6, podem servir de base para o cálculo de indíces de coerência organizacional.

Analogamente ao caso das relações de dependência organizacionais, a coerência organizacional pode também ser associada tanto ao nível da estrutura como ao nível da entidade organizacional, como mostrado a seguir.

6.1.7.1 Coerência da estrutura organizacional

A validação da coerência da estrutura organizacional é baseada em 2 (duas) noções:

 autonomia de um papel, que garante que para todas as missões associadas a este papel que devem delegar a ativação, execução ou utilização de objetivos, planos, ações, e recursos, existem necessariamente outros papéis na estrutura organizacional cujas missões permitem que tais delegações sejam aceitas; $6.1 \; Modelo \; \mathcal{M}{
m OISE}$

• coerência de um papel, que garante que um papel seja autônomo, e que a definição das relações e dos grupos da estrutura organizacional torne possível toda a delegação imposta pela estrutura organizacional.

Retomando mais uma vez o exemplo apresentado na tabela 6.1, observa-se que ambos os papéis são autônomos. Pode-se afirmar, por exemplo, que o papel « Candidato » é autônomo considerando o conjunto de papéis { « Candidato », « Secretária » } presentes na estrutura organizacional, pois este conjunto propicia ao papel « Candidato » todos os objetivos, planos, ações ou recursos que necessita para as suas missões.

Por outro lado, um papel é coerente se for autônomo e se qualquer objetivo, plano, ação ou recurso que não possa ser desempenhado pelo próprio papel possa ser delegado. No exemplo da tabela 6.1, o papel « Candidato » seria coerente caso a estrutura organizacional definisse uma relação de comunicação entre este papel e o papel « Secretária », no âmbito das missões envolvidas.

6.1.7.2 Coerência da entidade organizacional

A validação da coerência da entidade organizacional também é baseada em 2 (duas) noções:

- autonomia de um agente, que garante que todas as ações necessárias para que o agente desempenhe seu papel estejam disponíveis na entidade organizacional;
- coerência de atribuição de papéis, que garante que a distribuição de papéis é adequada em relação às capacidades dos agentes.

Novamente considerando o exemplo apresentado na tabela 6.1, suponhase uma vez mais que os agentes ag_1 e ag_2 desempenhem respectivamente os papéis de « Candidato » e de « Secretária ». Neste caso, pode-se dizer que ag_1 é autônomo, no desempenho do papel « Candidato » em relação ao conjunto de agentes $\{ag_1, ag_2\}$ presentes na entidade organizacional, pois este conjunto de agentes é capaz de realizar todas as ações que ag_1 necessita para desempenhar seu papel.

Suponha-se agora que o exercício de cada papel só possa ativar uma única missão por vez. Neste caso, a atribuição dos papéis acima não é coerente para ag_1 , pois não existe um agente capaz de realizar a ação « a_3 ». Caso outro agente também assumisse o papel de « Secretária », a coerência da atribuição de papéis estaria restabelecida.

A aplicação deste mecanismo de raciocínio organizacional, baseado nas noções de dependência e coerência organizacional, é ilustrada através da aplicação \mathcal{R} AISORG, descrita na seção 9.1.

6.2 Modelo \mathcal{M} oise⁺

Como o próprio nome diz, o modelo \mathcal{M} OISE⁺ (HÜBNER, 2003; HÜBNER; SICHMAN; BOISSIER, 2002a; HÜBNER; SICHMAN; BOISSIER, 2002c) consiste numa extensão do modelo \mathcal{M} OISE, descrito na seção anterior. Além de elucidar os componentes que formam uma organização em SMA e como estes se interrelacionam e contribuem para que os agentes do sistema alcancem seus objetivos, o modelo tem uma segunda finalidade. Ele foi desenvolvido para poder ser utilizado num contexto de reorganização, como discutido na seção 4.9. Em particular, o próprio modelo pode ser utilizado, de modo reflexivo, para propor uma organização de agentes, cujo objetivo é reorganizá-los para melhor atingir seus objetivos. O modelo apresenta, portanto, características que suportam tanto o projeto quanto a avaliação de novas organizações.

Como o modelo anterior que lhe serviu de inspiração, o modelo MOISE⁺, também diferencia as organizações abstratas das concretas, tal como discutido na seção 4.7, através da definição das noções de especificação organizacional e de entidade organizacional. A primeira representa ainda as restrições organizacionais claramente diferenciadas nos 3 (três) focos apresentados na seção 4.6: estrutural, funcional e deôntico.

Nas seções seguintes, os conceitos definidos acima são melhor detalhados. Em particular, será utilizado como exemplo ilustrativo destes conceitos num cenário de um time de futebol, que será também adotado na aplicação \mathcal{J} OJTEAM, descrita na seção 9.3.

6.2.1 Organização e autonomia dos agentes

Como explicado mais detalhadamente em (HÜBNER, 2003), os focos estrutural, funcional e deôntico do modelo $\mathcal{M}\text{OISE}^+$ são caracterizados pelas noções de papéis, metas globais e obrigações, que são detalhados respectivamente nas seções 6.2.2, 6.2.3 e 6.2.4. A relação entre estes focos e a forma como contribuem para facilitar que os agentes de um SMA atinjam seus objetivos é detalhada na figura 6.2. Nesta figura, considera-se que a finalidade da dimensão organizacional de um SMA é a de restringir o comportamento dos agentes, como discutido na seção 4.10. Na figura, B representa o espaço de comportamentos, e o conjunto P contém os comportamentos que conduzem os agentes aos seus objetivos. Resumidamente, um comportamento representa as ações e interações dos agentes do sistema².

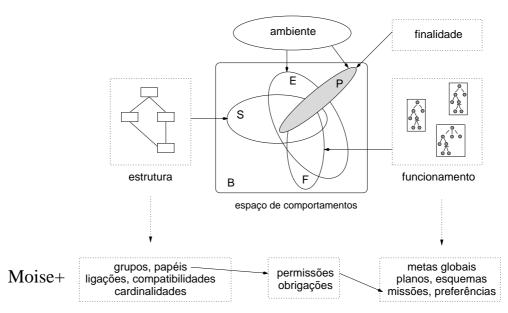


Figura 6.2: Efeito da organização no comportamento dos agentes (HÜBNER, 2003).

Na mesma figura, o conjunto E representa todos os comportamentos possíveis em um dado ambiente $(E \subset B)$. A estrutura organizacional restringe o comportamento dos agentes àqueles dentro do conjunto S $(S \subset B)$, e mais precisamente, cada agente (ou papel que desempenha) é relacionado a um sub-conjunto de S. Assim, os comportamentos possíveis para os agentes $(E \cap S)$ passam a estar mais próximos dos comportamentos desejados (P). Contudo, identificar P em $(E \cap S)$

²Uma descrição mais detalhada desta noção se encontra em (HÜBNER, 2003).

pode ser um problema nos casos onde esta intersecção represente um conjunto muito grande de comportamentos possíveis. Para reduzir esta busca, a especificação funcional contém um conjunto de planos globais F que têm se mostrado eficientes em conduzir o comportamento dos agentes de um ponto em $(E \cap S) - P$ para um ponto em $(E \cap S)$. Neste contexto, a especificação deôntica indica que papéis (em S) devem utilizar os comportamentos definidos em F.

Normalmente não se deseja especificar uma organização de forma muito rígida, onde os comportamentos possíveis são exatamente aqueles que conduzem os agentes a seus objetivos (S=P). Como P depende do ambiente (diferentes ambientes demandam diferentes comportamentos para se atingir os objetivos desejados), se S=P e o ambiente mudar (os comportamentos adequados passarem a ser P'), a organização pode deixar de ser eficaz $(S\cap P'=\varnothing)$. Ou seja, se S é muito restrito, o SMA pode ter problemas de adaptação a mudanças no ambiente, já que a autonomia dos agentes pode ter sido extinta. Por outro lado, se o conjunto S é muito amplo, a organização não é eficiente já que não restringe a autonomia dos agentes. Surge aqui, de forma mais clara, um dos problemas fundamentais no estudo das organizações dos agentes: $como\ conciliar\ a\ organização\ com\ a\ autonomia\ dos\ agentes$, isto é, como especificar uma boa organização.

A visão apresentada na figura 6.2 permite considerar o problema de encontrar uma boa organização (um bom tamanho para $S \cap F$) como um problema de busca em um espaço de comportamentos B, onde a organização dos agentes, composta de focos estruturais, funcionais e deônticos, pode ser considerada como uma heurística no processo de busca do comportamento almejado. Portanto, o sucesso na manutenção dos objetivos dos agentes em um SMA depende da sua organização e do ambiente. Como os agentes não têm controle total sobre o ambiente, uma forma de adaptá-los a uma eventual mudança no ambiente é alterar sua organização nos aspectos estruturais, funcionais e/ou deônticos. Este ponto será abordado com mais detalhes na seção 6.2.6.

6.2.2 Especificação estrutural

No foco estrutural, o modelo $\mathcal{M}OISE^+$ utiliza os mesmos conceitos definidos no modelo $\mathcal{M}OISE$: pap'eis, $rela\~{c}\~{o}es$ entre pap\'eis e grupos, descritos respectivamente nas sec $\~{o}$ es 6.1.2, 6.1.3 e 6.1.4.

O modelo \mathcal{M} OISE⁺, entretanto, propõe as seguintes extensões a estes conceitos:

- foi adicionada uma relação de herança entre papéis para simplificar a descrição da organização;
- foi adicionada uma relação de compatibilidade entre papéis para limitar os papéis que um agente pode assumir simultaneamente numa sociedade ou num grupo;
- foi adicionada a noção de cardinalidade e consequentemente o atributo de "bem formado" aos grupos;
- as ligações e a compatibilidade passam a ter um escopo (intra ou intergrupo);
- os grupos podem ter sub-grupos.

A figura 6.3 apresenta uma especificação estrutural de um time de futebol, utilizando o modelo \mathcal{M} OISE⁺.

Nesta figura, podem ser observadas algumas destas extensões, tais como: (i) um atacante ($\ll attacker \gg$) é um jogador ($\ll player \gg$) (herança); (ii) o capitão do time tem autoridade sobre os demais jogadores (relação de autoridade) e (iii) um time é formado por um grupo de ataque ($\ll attack \gg$), um grupo de defesa ($\ll defense \gg$), um capitão ($\ll leader \gg$) e um técnico ($\ll coach \gg$) (formação de grupos e sub-grupos).

6.2.3 Especificação funcional

Como discutido na seção 3.7, quando padrões e métodos de interação envolvendo os mesmos agentes se repetem várias vezes, estes podem ser capturados em estruturas pré-estabelecidas, diminuindo assim por exemplo a complexidade inerente dos métodos de divisão de tarefas e de coordenação. Tais estruturas normalmente têm um desempenho bastante eficiente, tanto pelo fato de terem sido aperfeiçoadas no decorrer do tempo quanto pelo fato de seus usuários terem se habituado a elas. Tais estruturas, que caracterizam o foco funcional da organização, são denominadas de esquemas sociais (LUGO; HÜBNER; SICHMAN, 2001).

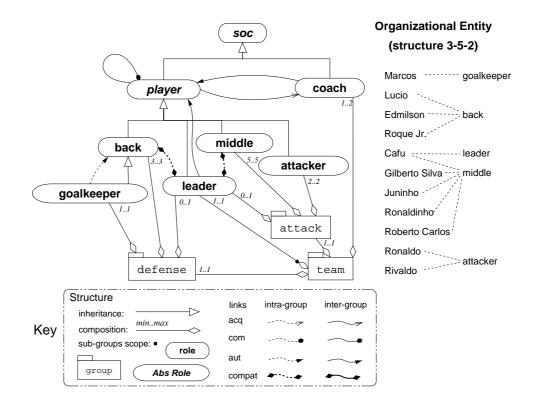


Figura 6.3: Foco estrutural de uma especificação e de uma entidade organizacional no modelo \mathcal{M} OISE⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c).

Um esquema social é composto por um conjunto de metas globais, estruturadas numa árvore de decomposição por meio de planos globais.

6.2.3.1 Metas globais

A noção fundamental para os esquemas sociais é a noção de *meta global*, que representa um estado do mundo que é desejado pelos agentes que compõem a organização.

Uma meta global tem normalmente associados 3 (três) parâmetros independentes que caracterizam seu estado:

- 1. *nível de satisfatibilidade*, que indica se esta meta já foi alcançada ou não, ou ainda se ela é impossível de ser alcançada;
- 2. *nível de alocação*, que indica se já existe ou não algum agente comprometido a satisfazê-la;

3. *nível de ativação*, que indica se as pré-condições necessárias para que a meta seja satisfeita estão presentes.

Inicialmente, uma meta global encontra-se num estado onde não está satisfeita, nem alocada e tampouco tem as suas pré-condições satisfeitas. À medida que o SMA evolui, o estado desta meta global se altera dinamicamente.

6.2.3.2 Missões

No nível individual, o modelo $\mathcal{M}\text{OISE}^+$ utiliza também o conceito de $miss\~ao$, definido no modelo $\mathcal{M}\text{OISE}$ na seção 6.1.2. Uma missão é um conjunto coerente de metas globais que pode ser atribuído a um agente através de um de seus papéis. O agente que se compromete com uma missão é responsável pela satisfação de todas as metas desta missão.

Como eventualmente uma meta global pode possuir mais de uma missão, vários agentes serão co-responsáveis pela sua satisfação. Na figura 6.4, apresenta-se um exemplo de especificação funcional de um time de futebol, utilizando o modelo \mathcal{M} OISE⁺.

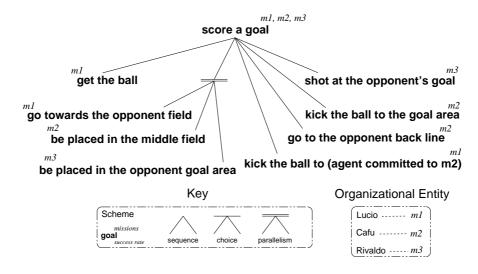


Figura 6.4: Foco funcional de uma especificação e de uma entidade organizacional no modelo \mathcal{M} OISE⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c).

Neste exemplo, a meta global principal do SMA é marcar um gol (« score a goal »), e existem diversas missões individuais, associadas às suas sub-metas, que devem ser realizadas pelos agentes, como por exemplo se posicionar na área do

 $6.2 \; Modelo \; \mathcal{M}{\rm OISE}^+$

adversário ("be placed in the opponent goal area", missão « m_3 ») e cruzar a bola para área ("kick the ball to the goal area", missão « m_2 »). Segundo o esquema social adotado, tais missões devem ser desempenhadas por agentes diferentes, e a primeira deve estar finalizada antes que a segunda possa ser iniciada. Verifica-se, deste modo, que a especificação funcional deste modelo na realidade implementa um método de coordenação de ações, conforme descrito na seção 3.5.4.

No que diz respeito ao foco funcional, o modelo \mathcal{M} OISE⁺ propõe as seguintes extensões ao modelo \mathcal{M} OISE:

- adicionou-se a noção de esquema social que permite ver os planos em um contexto social (global) e não no contexto de um papel como no modelo Moise. Em outras palavras, as missões passam a pertencer a um esquema social, e não mais a um papel;
- 2. foi adicionada a noção de cardinalidade e conseqüentemente o atributo de "bem formado" aos esquemas sociais;
- 3. pode-se determinar uma ordem de preferência entre as missões.

6.2.4 Especificação deôntica

Por fim, o aspecto deôntico liga os dois aspectos anteriores, indicando quais as responsabilidades dos papéis nos planos globais através das missões. A tabela 6.2 apresenta a especificação deôntica para o cenário do time de futebol. Pela tabela, pode-se observar que a jogada de ataque descrita na seção anterior, envolvendo as missões $\ll m_2 \gg e \ll m_3 \gg$, devem ser realizadas respectivamente por um membro do grupo do meio de campo ($\ll middle \gg$) e por um membro do grupo de ataque ($\ll attack \gg$).

Em relação ao foco deôntico, os seguintes aspectos do modelo \mathcal{M} OISE foram mais detalhados no modelo \mathcal{M} OISE⁺:

- 1. o foco deôntico passa a ter uma importância de primeiro nível, tal como os focos estrutural e funcional, pois no modelo Moise este aspecto ficava diluído no conceito de missão;
- 2. as relações deônticas passam a ter um tempo de validade.

 $6.2 \; Modelo \; \mathcal{M}{\rm OISE}^+$

Especificação Deôntica					
Papel	Relação Deôntica	Missão	Restrição Temporal		
<i>≪ back ≫</i>	permissão	$\ll m_1 \gg$	sempre		
$\ll middle \gg$	obrigação	$\ll m_2 \gg$	sempe		
$\ll attacker \gg$	obrigação	$\ll m_3 \gg$	sempre		

Tabela 6.2: Exemplo de especificação deôntica no modelo \mathcal{M} OISE⁺ (HÜBNER; SICHMAN; BOISSIER, 2002c).

6.2.5 Especificação e entidade organizacional

De modo similar ao modelo \mathcal{M} OISE, no modelo \mathcal{M} OISE⁺ são também diferenciadas as organizações abstratas das concretas, tal como discutido na seção 4.7, e representadas respectivamente nos conceitos de *especificação* e *entidade* organizacional.

No caso do modelo $\mathcal{M}OISE^+$, uma entidade organizacional é caracterizada pela associação de agentes a papéis (foco estrutural) e seu comprometimento com as missões que lhe cabem para atingir uma meta global (foco funcional). Nas figuras 6.3 e 6.4 tem-se alguns exemplos destas atribuições, como por exemplo o fato de Cafu assumir os papéis de capitão ("leader") e de jogador de meio campo ("middle"), assim como de se comprometer a realizar a missão $\ll m_2 \gg$.

6.2.6 Reorganização

O modelo \mathcal{M} OISE⁺ pode ser utilizado de modo reflexivo, para especificar uma organização cuja meta global seja reorganizar um certo grupo de agentes. A princípio, o modelo \mathcal{M} OISE⁺ pode oferecer inúmeras possibilidades para esta tarefa. Considerando o exemplo do time de futebol, cuja especificação organizacional é apresentada nas figuras 6.3, 6.4 e na tabela 6.2, podem-se imaginar as seguintes situações:

- substitui-se um jogador, por exemplo Ronaldo por Luizão (alteração da entidade organizacional);
- proibe-se que um jogador do meio de campo, como Ronaldinho, avance até a linha da grande área (alteração da especificação deôntica);

• altera-se o esquema de jogo, e tenta-se o ataque pela esquerda, com Roberto Carlos, ao invés de tentar o ataque pela direita, com Cafu (alteração da especificação funcional);

• coloca-se um jogador de defesa a mais, trocando um jogador de meio de campo por um quarto zagueiro, por exemplo Juninho por Anderson Polga (alteração da especificação estrutural).

No processo de reorganização proposto em (HÜBNER, 2003), apenas a primeira opção acima não é considerada. Retomando a discussão apresentada na seção 4.9, este processo adota a seguinte filosofia:

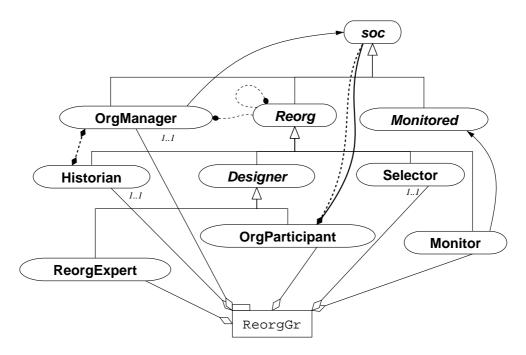


Figura 6.5: Especificação estrutural do processo de reorganização no modelo \mathcal{M} OISE⁺ (HÜBNER, 2003).

- 1. o processo de reorganização é limitado à especificação organizacional (organização abstrata);
- 2. o processo de organização é controlado, e consta das etapas de monitoração, projeto, seleção e implementação;
- 3. as etapas são realizadas de modo endógeno e distribuído;

 $6.2 \; Modelo \; \mathcal{M}{\rm OISE}^+$

4. os agentes que participam deste processo têm uma representação explícita do modelo organizacional.

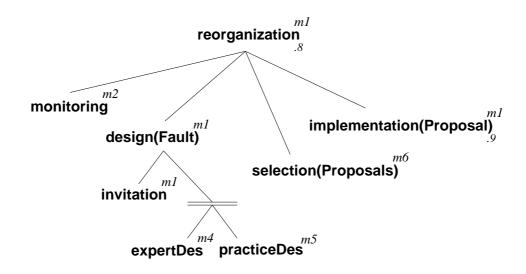


Figura 6.6: Especificação funcional do processo de reorganização no modelo \mathcal{M} OISE⁺ (HÜBNER, 2003).

As figuras 6.5, 6.6 e a tabela 6.3 apresentam respectivamente a especificação estrutural, funcional e deôntica do processo de reorganização, através do modelo \mathcal{M} OISE⁺.

Mesmo que existam algumas regras gerais, normalmente as etapas do processo de reorganização são dependentes do domínio de aplicação. Por exemplo, na etapa de monitoração, a detecção de falhas na organização depende de parâmetros e métodos específicos do domínio. No modelo $\mathcal{M}\text{OISE}^+$, estas particularidades são resolvidas por agentes do domínio de aplicação que assumem os papéis do processo de reorganização, mostrados na figura 6.5. No caso da etapa de seleção das propostas de reorganização, são apresentadas em (HÜBNER, 2003) duas alternativas: votação e aprendizado. Na primeira, abordada na seção 3.5.5, os agentes votam nas propostas com base na reputação dos seus propositores. Na segunda alternativa, o sistema aprende quais são as propostas que têm maior chance de sucesso caso implementadas. Estas duas soluções também não são universais, porém podem ser aplicadas em vários outros domínios. Esta segunda alternativa foi adotada no sistema \mathcal{J} OJTEAM, descrito na seção 9.3.

 $6.3 \quad Conclus\~oes$

Especificação Deôntica						
P	RD	M	RT			
$<\!\!< OrgManager >\!\!>$	obrigação	$m_1 = \{ \ll reorganization \gg, \ \ \ \ \ \ \ \ \ \ \ \ \ $	sempre			
$\ll Monitor \gg$	obrigação	$m_2 = \{ \ll monitoring \gg \}$	sempre			
$\ll ReorgExpert \gg$	obrigação	$m_4 = \{ \ll expertDes \gg \}$	sempre			
$\ll OrgParticipant \gg$	permissão	$m_5 = \{ \ll practiceDes \gg \}$	sempe			
$\ll Selector \gg$	obrigação	$m_6 = \{ \ll selection \gg \}$	sempre			

onde: P = papel, RD = relação deôntica, M = missão, RT = restrição temporal

Tabela 6.3: Especificação deôntica do processo de reorganização no modelo \mathcal{M} OISE⁺ (HÜBNER, 2003).

6.3 Conclusões

A figura 6.7 mostra os modelos de raciocínio organizacional \mathcal{M} OISE e \mathcal{M} OISE⁺ classificados segundo os eixos básicos propostos na seção 2.3.

Em relação à sua finalidade, ambos os modelos podem ser utilizados tanto numa perspectiva de resolução social, como numa perspectiva de simulação social. Esta afirmação pode ser comprovada através das aplicações Collorg (resolução social, modelo Moise), Raisorg (simulação social, modelo Moise) e Jojteam (resolução social, modelo Moise), descritas no capítulo 9. Os dois modelos se situam no nível de descrição teórico. Finalmente, ambos os modelos abordam a dimensão organizacional da metodologia Vogais.

Considerando os aspectos referentes às características de agentes e SMA, abordados no capítulo 2, ambos os modelos pressupõem que os agentes que os utilizam sejam autônomos, cognitivos e sociais, conforme a taxonomia apresentada na seção 2.1.2, e cuja arquitetura seja deliberativa ou híbrida, conforme discutido na seção 2.1.6. Segundo ainda a taxonomia de SMA proposta na seção 2.2.2, os modelos podem ser aplicados em SMA abertos ou não, de baixa granularidade (poucos agentes), compostos por agentes homogêneos ou heterogêneos. Obviamente, neste segundo caso os agentes devem ter associados a si o mecanismo de raciocínio organizacional proposto pelos modelos.

 $6.3 \quad Conclus\~oes$ 124

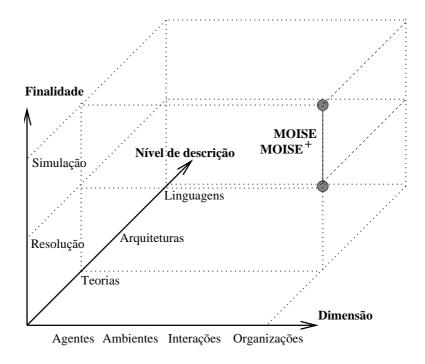


Figura 6.7: Classificação dos modelos de raciocínio organizacional.

No que diz respeito à discussão sobre a interação entre agentes, apresentada no capítulo 3, os modelos foram propostos para explorar e estruturar situações de interação cuja interferência social seja positiva, tais como a colaboração simples ou coordenada, discutidas na seção 3.2.2. As noções de missão (modelos \mathcal{M} OISE e \mathcal{M} OISE⁺) e meta global (modelo \mathcal{M} OISE⁺) têm o objetivo de pré-definir alguns métodos de interação descritos na seção 3.5, como divisão de tarefas, planejamento e coordenação de ações. Finalmente, a noção de relação organizacional (modelo \mathcal{M} OISE) restringe a linguagem e os protocolos de comunicação, discutidos nas seções 3.6.2 e 3.6.4, utilizados pelos agentes que desempenham os papéis presentes na estrutura organizacional.

Considerando ainda as propriedades de organizações de agentes, discutidas no capítulo 4, ambos os modelos abordam os focos estrutural, funcional e deôntico presentes em uma organização, conforme a taxonomia apresentada na seção 4.4. Em ambos os modelos, os focos estrutural e funcional são caracterizados respectivamente pelos conceitos de papel e de missão. No modelo \mathcal{M} OISE, o foco deôntico é representada pela natureza da missão, enquanto que no modelo \mathcal{M} OISE⁺ o foco deôntico tem a mesma importância dos outros dois. Ainda no modelo \mathcal{M} OISE⁺, o foco funcional é caracterizado também pelo conceito de meta global, ausente

 $6.3 \ Conclus\~oes$

do modelo \mathcal{M} OISE. Quanto à sua gênese, tal como vista na seção 4.5, ambos os modelos consideram um modelo de organização formal, pré-existente à atividade do SMA. Em ambos os modelos representam-se os níveis abstrato e concreto das organizações, conforme discutido na seção 4.7, respectivamente através dos conceitos de estrutura e entidade organizacional. Quanto ao comportamento das organizações, como visto na seção 4.9, o modelo \mathcal{M} OISE pressupõe um modelo de organização estático³, enquanto que o modelo \mathcal{M} OISE⁺ permite melhor abordar o caráter dinâmico das organizações, já que foi explicitamente concebido para facilitar o processo de reorganização.

Finalmente, no que diz respeito à representação das interações e das organizações, discutidas nas seções 3.3 e 4.8, ambos os modelos podem ser utilizados como parte da arquitetura de agentes cuja representação organizacional seja explícita, numa perspectiva de resolução de problemas. Por outro lado, nada impede que os modelos sejam utilizados como ferramenta de projeto de um SMA, auxiliando o projetista a restringir as interações entre os agentes do sistema, em cuja situação a representação organizacional seria implícita. Estas duas abordagens seriam representadas respectivamente pelos casos (d) e (c) mostrados na figura 4.4.

³Na verdade, este modelo não leva em conta o fenômeno da reorganização em sua formulação.

Parte III

Aplicações

7 Ambientes de Desenvolvimento

Devido à inerente complexidade envolvida no desenvolvimento de SMA, normalmente utilizam-se ambientes de desenvolvimento que possam, ao menos em parte, liberar o projetista de certas tarefas, como por exemplo a comunicação entre agentes, a distribuição transparente em uma rede de máquinas, a customização de uma sessão de simulação, entre outras.

O objetivo deste capitulo é apresentar alguns ambientes que foram desenvolvidos (Masenv, Saci) ou estão em desenvolvimento (Simcog) com esta finalidade. Tais ambientes são detalhados respectivamente nas seções 7.1, 7.2 e 7.3. Adicionalmente, pequenos trechos de código de agentes desenvolvidos nos ambientes Masenv e Saci encontram-se descritos respectivamente nos apêndices I e II. Na seção 7.4, estes ambientes são situados em relação à classificação proposta na seção 2.3.

Mais detalhes sobre os ambientes \mathcal{M} ASENV, \mathcal{S} ACI e \mathcal{S} IMCOG podem ser encontrados respectivamente em (CARDOZO; SICHMAN; DEMAZEAU, 1993; SICHMAN, 1995; SICHMAN, 1997a), (SACI, 2000; HÜBNER; SICHMAN, 2000a; ALBUQUERQUE et al., 2002) e (MARIETTO et al., 2002; SIMCOG, 2002).

7.1 Ambiente \mathcal{M} asenv

O ambiente Masenv (Multi-Agent Systems Environment) (CARDOZO; SICH-MAN; DEMAZEAU, 1993; SICHMAN, 1995; SICHMAN, 1997a) foi concebido para prover a um projetista de SMA algumas funcionalidades que pudessem auxiliá-lo a prototipar de modo mais rápido a sua aplicação, tais como protocolos de interação, modelos organizacionais, bibliotecas de agentes e ambientes simulados.

O ambiente foi concebido de acordo com os elementos básicos da metodologia Vogais, descrita na seção 2.3.3. Nesta seção, será descrita exclusivamente a parte do ambiente responsável pela comunicação entre agentes.

No desenvolvimento deste ambiente, considerou-se que o modelo de objetos ativos, descrito na seção 2.2.5 pudesse ser utilizado como substrato para a implementação de SMA.

7.1.1 Implementando agentes como objetos ativos

Seja então um modelo genérico de agente, que além das capacidades de raciocínio e decisão presentes nos sistemas baseados em conhecimento usuais, tem as capacidades discutidas na seção 2.1.1, tais como percepção, comunicação, e ação. Adicionalmente, para que pudesse operar num ambiente aberto, deveria ser dotado de um mecanismo de raciocínio social que lhe permitisse raciocinar sobre os outros agentes, representados numa descrição externa, como descrita na seção 5.1.4. Finalmente, uma estrutura de controle complexa deveria ser posta em prática para ativar cada um destes tratamentos no momento adequado.

Este modelo genérico de agente pode ser implementado como um objeto ativo:

- o estado público do objeto pode ser o local destinado a armazenar a descrição externa do agente, bem como informações a serem trocadas assincronamente entre os agentes para efeito de comunicação e percepção;
- o estado privado pode ser utilizado para comunicação entre os módulos internos do agente já que sendo uma entidade complexa, um agente pode por sua vez também ser implementado como um conjunto de "threads" de controle;
- comunicação e percepção podem ser efetuadas através da ativação de métodos definidos para estes fins;
- agentes podem definir métodos adicionais dependentes de domínio para auxiliar na resolução de conflitos, controlar protocolos de interação, etc.

A figura 7.1 mostra como o modelo genérico de agente pode ser implementado como um objeto ativo. Cabe ao projetista definir se o agente deve conter um ou

mais "threads" de controle. Caso sejam utilizados diversos "threads" de controle, as atividades internas do agente podem ser realizadas em paralelo, necessitando para isto um controle mais elaborado.

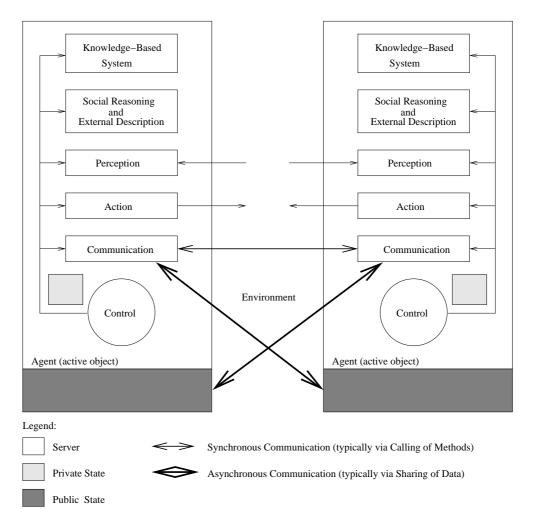


Figura 7.1: Implementação do modelo de agente utilizando objetos ativos (CARDOZO; SICHMAN; DEMAZEAU, 1993).

7.1.2 SISTEMA \mathcal{D} PSK+P

O sistema $\mathcal{D}PSK+P$ (Distributed Problem Solving Kernel) (CARDOZO; SICH-MAN, 1992) permite o desenvolvimento de aplicações baseadas no modelo de objetos ativos, cujos servidores são codificados nas linguagens C++ e CLOS. Cabe ressaltar que tal sistema foi um dos pioneiros no país na utilização de objetos distribuídos, já que foi desenvolvido no início da década de 90.

Do ponto de vista do usuário, o sistema diferia de outros sistemas então dis-

poníveis pelo modo de tratar a distribuição dos objetos, bem como a sua transparência. Ao invés de extender as linguagens de programação, o sistema fornece duas bibliotecas de classes, uma para servidores baseados em C++ e outra para servidores baseados em CLOS. Toda a complexidade relativa ao processamento distribuído, como a comunicação e controle entre processos, o gerenciamento de dados compartilhados e o serviço de nomes, é tratada nestas classes e ocultada do usuário. Assim, este último pode criar aplicações distribuídas a partir das linguagens usuais, sem a necessidade de utilização de novas construções (e compiladores associados).

O sistema, que executa no ambiente UNIX, tem basicamente 3 (três) componentes:

- um núcleo, responsável por funções como comunicação entre máquinas, manutenção de dados compartilhados, gerenciamento de processos, e serviço de nomes;
- 2. uma *interface*, composta por 2 (duas) duas bibliotecas de classes (APIs), que contêm métodos responsáveis pelo processamento dos objetos ativos;
- 3. um sistema de apoio à execução ("run-time support system"), que contém um conjunto de ferramentas de monitoração para gerenciar e inspecionar os objetos ativos (SICHMAN, 1991; SICHMAN; CARDOZO, 1992a; SICHMAN; CARDOZO, 1992b).

O gerenciamento dos objetos ativos é composto das seguintes atividades:

- construção de objetos compartilhados, tanto classes como instâncias;
- manipulação de objetos ativos, tais como acesso, atualização e destruição de instâncias;
- processamento de transações, para disciplinar o acesso aos objetos ativos, de modo a garantir as propriedades de consistência e persistência;
- processamento de métodos, que utiliza um protocolo que se situaria entre uma RPC, já que servidores tratam as chamadas, e um Rendez-Vous (RDV), já que as chamadas devem ser explicitamente aceitas por um servidor;

• controle de servidores, já que os servidores associados a objetos podem ser inicializados, suspensos, reiniciados, interrompidos ou finalizados através do envio explícito de sinais para os objetos.

As ferramentas de monitoração permitem que o usuário possa obter 3 (três) tipos de informações distintas:

- verificar os eventos de controle que alteram o estado de execução dos servidores de objetos;
- verificar as interações que modificam o estado público dos objetos ativos;
- verificar o conteúdo da memória compartilhada, em três níveis distintos: hierarquia de classes, protótipo de classe e estado de um determinado objeto ativo, como mostra a figura 7.2.

Sempre que dois objetos ativos interagem, o sistema de apoio à execução gera um *evento* que captura as informações referentes a esta interação. Caso as ferramentas de monitoração estejam ativadas, tais eventos lhes são então encaminhados, para que possam atualizar o estado da aplicação.

7.1.3 Comunicação entre agentes no ambiente \mathcal{M} asenv

Através da utilização do sistema $\mathcal{D}PSK+P$, foram projetadas duas classes compartilhadas, Mbox e Message, para implementar o mecanismo de caixas postais. Mensagens trocadas entre os agentes são objetos ativos derivados da classe Message.

Certas facetas definidas na classe *Mbox* armazenam elementos de indentificação do agente, tais como seu nome simbólico, máquina onde está executando e identificador de processo. Outras facetas apontam para as mensagens enviadas ao agente, conforme a figura 7.2.

Na classe *Mbox*, definem-se alguns métodos tais como *Send*, *Receive* e *Number_of_messages*, respectivamente utilizados para o envio e recebimento de mensagens, e para obter o número de mensagens pendentes na caixa postal do agente. As mensagens trocadas entre os agentes são implementadas como objetos ativos derivados da classe *Message*.

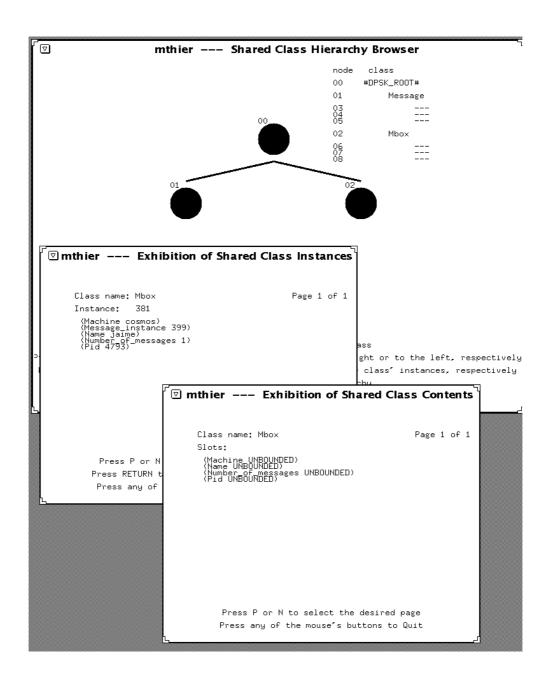


Figura 7.2: Monitoração do conteúdo da memória compartilhada no sistema \$\mathcal{D}_{PSK+P}\$ (CARDOZO; SICHMAN; DEMAZEAU, 1993)

Um pequeno exemplo de código de agente no ambiente \mathcal{M} ASENV, que ilustra a utilização do ambiente, pode ser encontrado no apêndice I.

7.2 Ambiente \mathcal{S}_{ACI}

A implementação de alguns tipos de SMA requer que os agentes estejam distribuídos numa rede de computadores. Apesar da grande variedade de ferramentas disponíveis para auxiliar nesta tarefa, várias dificuldades são encontradas na sua utilização, tais como: (i) baixa interoperabilidade, pois geralmente uma única linguagem de programação é suportada ou a ferramenta pode ser utilizada em um único sistema operacional; (ii) baixo desempenho, fato que inviabiliza sua utilização em sistemas onde há comunicação intensa entre os agentes; (iii) difícil utilização, pois a instalação, configuração e utilização destas ferramentas complexas requerem do programador conhecimento de programação distribuída, protocolos de rede, etc.; e (iv) falta de robustez, pois na maioria dos casos o ambiente não funciona adequadamente por muito tempo.

Tais dificuldades motivaram o desenvolvimento do ambiente Saci (Simple Agent Communication Infrastructure) (Saci, 2000; Hübner; Sichman, 2000a; Albuquerque et al., 2002). Trata-se de uma ferramenta que torna transparente para o programador de agentes os aspectos de comunicação distribuída, tendo como principais objetivos a simplicidade de utilização e o bom desempenho.

O ambiente Saci foi desenvolvido na linguagem Java, e utiliza a linguagem KQML (FININ et al., 1994), apresentada na seção 3.6.2, como linguagem externa de comunicação entre agentes. O ambiente possui as seguintes características principais:

- provê métodos para que os agentes possam compor, enviar e receber mensagens na linguagem KQML;
- identifica os agentes por um nome, e as mensagens são transportadas utilizandose somente o nome do receptor, sendo sua localização na rede transparente;
- agentes podem conhecer os outros por meio de um serviço de *páginas bran*cas ("Agent Name Server");

• agentes podem ser implementados como "applets" e terem sua interface em uma "home-page";

- agentes podem ser iniciados remotamente;
- monitoramento da atuação dos agentes, através da visualização e armazenamento de eventos sociais, tais como entrada e saída de agentes no sistema, recebimento ou envio de mensagens.

7.2.1 Entrada e saída de agentes

No modelo do ambiente SACI, agentes são agrupados em sociedades. Em cada sociedade existe um agente especial, denominado facilitador, cuja função é armazenar a identidade e localização dos agentes desta sociedade, bem como os serviços por eles oferecidos.

Quando um agente deseja entrar numa determinada sociedade, ele deve contatar o respectivo facilitador e registrar um nome. O facilitador irá verificar a unicidade deste nome e associá-lo à localização do agente. Da mesma forma, quando deseja sair da sociedade, o agente deve também avisar o facilitador.

7.2.2 Envio e recebimento de mensagens

De modo semelhante ao ambiente \mathcal{M} ASENV, descrito na seção 7.1, o ambiente \mathcal{S} ACI possui um componente denominado MBox , que serve de interface entre o agente e a sociedade. Sua finalidade é tornar transparente o envio e o recebimento de mensagens. Este componente possui funções que encapsulam a composição de mensagens KQML, o envio síncrono e assíncrono de mensagens, o recebimento de mensagens, o anúncio e a consulta de habilidades e a difusão ("broadcast") de mensagens.

7.2.3 ANÚNCIO DE HABILIDADES

Para serem melhor conhecidos na sociedade, os agentes podem anunciar suas habilidades ao facilitador. Assim, quando um agente necessita de um serviço e não conhece um agente capaz de realizá-lo, pode requisitar ao facilitador uma lista de agentes com tal habilidade. A figura 7.3 ilustra um exemplo dessa interação.

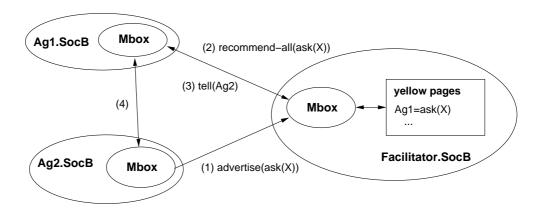


Figura 7.3: Exemplo de anúncio de habilidades no ambiente \mathcal{S} ACI (HÜBNER; SICHMAN, 2000a).

O agente Ag2 da sociedade SocB anuncia ao facilitador que é capaz de realizar o serviço X (1). O agente Ag1 desta sociedade, num dado momento, necessita deste serviço, e pergunta ao facilitador quem pode provê-lo na sociedade (2). Obtendo como resposta Ag2 (3), ele toma a iniciativa de contatar diretamente este agente (4), para obter o serviço que deseja.

7.2.4 Exemplo de interação no ambiente \mathcal{S} aci

A figura 7.4 ilustra uma interação mais elaborada entre dois agentes, correspondente a um serviço de $p\'{a}ginas\ brancas$.

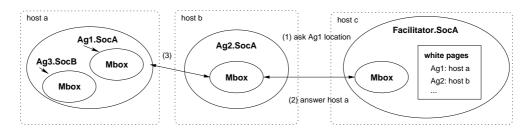


Figura 7.4: Serviço de páginas brancas no ambiente SACI (HÜBNER; SICHMAN, 2000a).

O agente Ag2 da sociedade SocA deseja se comunicar com o agente Ag1 que pertence à mesma sociedade. Inicialmente, o componente MBox de Ag2 precisa saber a localização de Ag1, e portanto pergunta ao facilitador da sociedade SocA qual é esta localização (1). O facilitador lhe responde host a (2). Tendo a localização, o Ag2 inicia a comunicação com o agente Ag1 (3). O agente que

se encontra no host a pertence a duas sociedades SocA e SocB, tendo um Mbox associado para cada sociedade.

7.2.5 Ambiente \mathcal{K} Saci

Uma outra tendência no processamento de informações neste início de século XXI é a chamada computação pervasiva ("pervasive computing"). De acordo com esta tendência, serviços e informações serão disponibilizados aos usuários independentemente de horário e local ("anytime, anywhere services"), através da utilização intensiva da capacidade de processamento de pequenos dispositivos móveis e conectados, como telefones celulares e computadores de mão ("palmtops").

Um esforço recente no sentido de viabilizar a computação pervasiva é a especificação e desenvolvimento da linguagem J2ME (Java 2 Micro Edition, também conhecida como KJava) (SUN, 2000, apud ALBUQUERQUE et al. 2001). Esta linguagem é um subconjunto da versão "standard" da linguagem Java, onde grande parte das bibliotecas do ambiente de execução foram retiradas para permitir que pudesse ser instalada em pequenos dispositivos móveis, com capacidade de memória limitada.

O ambiente KSACI (ALBUQUERQUE et al., 2002) é uma extensão do ambiente SACI, destinada a possibilitar um modelo de computação pervasiva, onde agentes podem ser executados em dispositivos móveis. Resumidamente, dada uma sociedade de agentes, o seu facilitador é executado numa máquina "desktop", no ambiente SACI, e os agentes da sociedade podem ser de 3 (três) tipos distintos: (i) agentes SACI, que são executados numa máquina "desktop", (ii) agentes KSACI, que são executados em dispositivos móveis e (iii) agentes externos, desenvolvidos em outros ambientes. Para os agentes KSACI, é criado um representante ("proxy") na máquina "desktop" que possibilita o contato com o facilitador¹. Uma política similar é utilizada para os agentes externos. A figura 7.5 mostra um exemplo de interação de agentes que utilizam o ambiente KSACI.

Um pequeno exemplo de código de agente no ambiente \mathcal{S} ACI, que ilustra a utilização do ambiente, pode ser encontrado no apêndice II.

 $^{^1}$ Isto é necessário pois a linguagem J2ME não permite a utilização do protocolo RMI, utilizado no ambiente $\mathcal{S}_{\rm ACI}.$

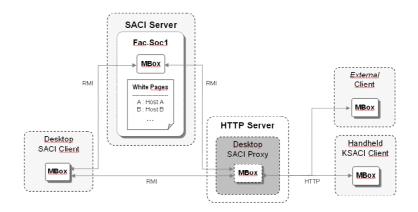


Figura 7.5: Exemplo de interação entre agentes no ambiente KSACI (ALBUQUERQUE et al., 2002).

7.3 Ambiente Simcog

Apesar de haver um interesse cada vez mais crescente na utilização de SMA para realizar simulações sociais (CONTE; SICHMAN; GILBERT, 1998), conforme discutido na seção 2.3.1, a maior parte das plataformas desenvolvidas até o momento para este fim privilegiam os SMA reativos (SWARM, 1999; LE PAGE et al., 2000).

O ambiente Simcog (Simulation of Cognitive Agents) (MARIETTO et al., 2002; SIMCOG, 2002) está sendo concebido como uma ferramenta flexível cujo objetivo é auxiliar os pesquisadores nas áreas de ciências sociais na simulação computacional dos fenômenos que lhes interessam. Mais especificamente, o ambiente Simcog funcionará como uma ferramenta de análise do modelo conceitual, auxiliando de forma mais objetiva a observação de fenômenos e resultados emergentes não previstos no modelo teórico-formal proposto.

Em termos de especificação de requisitos, realizou-se um extenso levantamento das necessidades da área, detectando-se que uma tal plataforma deveria prover uma série de serviços, classificados em 5 (cinco) categorias:

- 1. facilidades tecnológicas, que intermediam a interação da plataforma com o sistema operacional e os serviços de rede, caso sejam necessárias várias máquinas para efetuar simulações de larga escala;
- 2. facilidades de domínio, que provêm elementos que permitam impor certas características do ambiente multiagentes a ser simulado, como por exemplo

7.4 Conclusões

a possibilidade de gerar falhas intencionais e controladas de comunicação;

3. facilidades de desenvolvimento, que possam por exemplo fornecer ao usuário modelos de agentes e modelos organizacionais para que este possa prototipar mais rapidamente a sua simulação, utilizando a mesma filosofia da metodologia Vogais, apresentada na seção 2.3.3;

- 4. facilidades de análise, que auxiliem o usuário a coletar e analisar os resultados da simulação, através da integração de pacotes estatísticos e de ferramentas que auxiliem a descobrir padrões de emergência de ações coletivas;
- 5. facilidades de exploração, que englobam serviços que tratam da exploração de diferentes resultados e conceitos qualitativos emergentes.

A análise de requisitos realizada foi validada e aperfeiçoada através de um questionário "on-line" enviado à comunidade científica da área de simulação multiagentes. Ao final deste processo, o questionário havia sido respondido por 196 (cento e noventa e seis) pesquisadores da área.

Algumas destas categorias de serviços, como as facilidades tecnológicas, já fazem parte de muitos ambientes de desenvolvimento para SMA, como por exemplo os ambientes \mathcal{M} ASENV e \mathcal{K} SACI, descritos anteriormente neste capítulo. Outras, como as facilidades de análise, e principalmente de exploração, são mais específicas da finalidade de simulação social. Nesta última reside a principal contribuição que tal ambiente pretende oferecer aos pesquisadores que o utilizarem no futuro. Este ambiente encontra-se atualmente em fase de desenvolvimento.

7.4 Conclusões

A figura 7.6 mostra os ambientes de desenvolvimento \mathcal{M} ASENV, \mathcal{S} ACI e \mathcal{S} IMCOG classificados segundo os eixos básicos propostos na seção 2.3.

Em relação à sua finalidade, os ambientes \mathcal{M} ASENV e \mathcal{S} ACI foram desenvolvidos como suporte a aplicações com a perspectiva de resolução social. Já por sua vez, o ambiente \mathcal{S} IMCOG visa ser utilizado num contexto de simulação social. Os três ambientes se situam no nível de linguagens. Finalmente, enquanto o objetivo

7.4 Conclusões 139

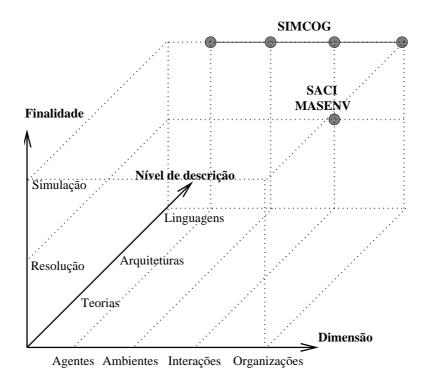


Figura 7.6: Classificação dos ambientes de desenvolvimento.

do ambiente $\mathcal{S}_{\text{IMCOG}}$ é o de abordar todas as dimensões presentes na metodologia Vogais, os ambientes $\mathcal{M}_{\text{ASENV}}$ e \mathcal{S}_{ACI} se limitam à dimensão da interação entre agentes, mais particularmente no que diz respeito à sua comunicação, como abordado na seção 3.6.

No que diz respeito aos modelos de implementação, o ambiente \mathcal{M} ASENV utiliza o modelo de objetos ativos, enquanto que o ambiente \mathcal{S} ACI utiliza o modelo SOD, apresentados na seção 2.2.5.

8 Aplicações em Raciocínio Social

Neste capítulo, são apresentadas respectivamente nas seções 8.1, 8.2 e 8.3 as aplicações \mathcal{D} EPNET, \mathcal{D} EPINT, e \mathcal{D} EPINT⁺. Tais aplicações são descritas de forma resumida, de modo a realçar suas características principais. Na seção 8.4, estas aplicações são situadas em relação à classificação proposta na seção 2.3.

Mais detalhes sobre as aplicações \mathcal{D} EPNET, \mathcal{D} EPINT, e \mathcal{D} EPINT⁺ podem ser encontrados respectivamente em (CONTE; SICHMAN, 1995; SICHMAN, 1995), (SICHMAN, 1995; SICHMAN, 1998) e (DAVID, 1998; DAVID; SICHMAN; COELHO, 1999; DAVID; SICHMAN; COELHO, 2000).

8.1 Aplicação \mathcal{D} epnet

 \mathcal{D} EPNET (SICHMAN, 1995; CONTE; SICHMAN, 1995) é um simulador concebido para ilustrar a utilização do modelo \mathcal{S}_{RM} , descrito na seção 5.1, num contexto de simulação social, como discutido na seção 2.3.1. Através do simulador, podem-se criar sociedades de agentes, calcular as suas relações e situações de dependência e construir as redes de dependência correspondentes.

Neste simulador, existe uma única visão objetiva das relações de dependência, e portanto existe apenas uma única descrição externa, conforme apresentado na seção 5.1.4, que é compartilhada por todos os agentes.

O simulador é composto dos seguintes módulos:

1. edição da descrição externa, que permite que o usuário crie novos agentes, editando seus objetivos, planos, ações e recursos, ou que modifique as informações referentes a um agente já existente;

****** DEPENDENCE NETWORK SIMULATOR ******

Jaime Sichman LIFIA/IMAG, Grenoble, France IP/CNR, Roma, Italia

DEPENDENCE SITUATIONS MENU

- Type 1 (Calculate mutually believed mutual dependence)
 - 2 (Calculate locally believed mutual dependence)
 - 3 (Calculate mutually believed reciprocal dependence)
 - 4 (Calculate locally believed reciprocal dependence)
 - 5 (Calculate unilateral dependence)
 - 6 (Print the external description)
 - 7 (Return to dependence network menu)
 - 8 (Return to main menu)
 - 9 (Quit)

Figura 8.1: Exemplo de interface do simulador Depnet (CONTE; SICHMAN, 1995).

- 2. construção de redes de dependência, relativas a um agente especificado pelo usuário, e parametrizadas pelo objetivo (redes completas envolvendo todos os objetivos ou redes específicas para um determinado objetivo) e pelo tipo (redes de dependência de ações ou de recursos);
- 3. cálculo de situações de dependência, relativas a um agente e objetivo, sendo que o usuário deve especificar qual tipo de situação ele está interessado em analisar;
- 4. cálculo de situações de objetivo, relativas a um dado agente e objetivo especificados pelo usuário;
- 5. cálculo de planos factíveis e objetivos realizáveis, relativos a um dado objetivo e plano especificados pelo usuário;
- 6. gestão do disco, que permite salvar e recuperar do disco as sessões de simulação realizadas.

O simulador Depnet é executado em plataformas UNIX, e foi desenvolvido

na linguagem C++, tendo aproximadamente 5800 linhas de código. Sua interface para o cálculo de situações de dependência é apresentada na figura 8.1.

Como um exemplo para ilustrar o cálculo das situações de dependência, seja uma sociedade composta por 7 (sete) agentes *jaime*, rosaria, cristiano, vittorio, maria, amedeo e paola, cuja descrição externa é exibida pelo simulador conforme abaixo:

```
name:
            jaime
                                              name:
                                                          rosaria
            g1 g2 g3 g4
goals:
                                              goals:
                                                          g1
actions:
            a1
                                              actions:
                                                          a2
                                              resources: r2
resources: r1
plans:
            g1:=a3(r1).
                                              plans:
                                                          g1:=a2(r2).
            g2:=a6(r1), a7(r2).
            g3:=a1(r1), a2(r3).
            g4:=a1(r1), a4(r5), a5(r4).
name:
            cristiano
                                              name:
                                                          vittorio
                                                          g4
            g5
goals:
                                              goals:
actions:
            a3
                                              actions:
                                                          a4
resources: r3
                                              resources: r4
                                                          g4:=a4(r5), a5(r4).
            g5:=a3(r3).
plans:
                                              plans:
name:
            maria
                                              name:
                                                          amedeo
goals:
            g4
                                              goals:
                                                          g3
actions:
                                              actions:
            a5
                                                          a6
resources: r5
                                              resources: r6
            g4:=a1(r1), a5(r7).
                                                          g3:=a6(r6).
plans:
                                              plans:
name:
            paola
goals:
            g3
actions:
            a7
resources: r7
```

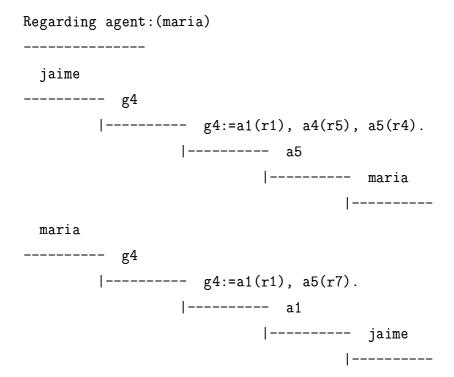
Utilizando a interface para o cálculo de situações de dependência, podem-se

g3:=a1(r2), a7(r3).

plans:

obter por exemplo os seguintes resultados:

(1) existe uma mutual believed mutual dependence entre os agentes jaime e maria relativo a g_4 , conforme baixo:



(2) o agente jaime calcula uma locally believed reciprocal dependence entre ele e o agente amedeo para o objetivo g_2 , conforme abaixo:

8.2 Aplicação \mathcal{D} epint

 \mathcal{D} epint (SICHMAN, 1995; SICHMAN, 1998) é um sistema concebido para ilustrar a utilização do modelo \mathcal{S}_{RM} , descrito na seção 5.1, num contexto de resolução social, como discutido na seção 2.3.1. O sistema implementa a técnica CBD, descrita na seção 5.1.11.

Diferentemente do simulador \mathcal{D} EPNET, descrito na seção 8.1, cada agente do sistema mantém a sua própria descrição externa, conforme descrita na seção 5.1.4.

8.2.1 Objetivos do sistema

Os objetivos do sistema \mathcal{D} EPINT são os seguintes:

- 1. mostrar que a utilização do modelo SRM permite melhorar a adaptação de um agente a SMAs abertos, através da utilização das noções de plano factível e objetivo atingível, apresentadas na seção 5.1.11, de modo garantir que todas as ações necessárias à execução de um plano estejam disponíveis quando um agente o escolher para atingir um determinado objetivo (SICH-MAN, 1995; SICHMAN, 1997b);
- 2. mostrar que a utilização da noção de situação de dependência, descrita na seção 5.1.10, como medida de avaliação da susceptibilidade de agentes adotarem objetivos uns dos outros, pode melhorar a eficiência do processo de formação de coalizões (SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995b);
- auxiliar a manter a consistência das informações que os agentes têm uns sobre os outros, através de um mecanismo de revisão distribuída de crenças (SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995a; SICHMAN; DEMAZEAU, 1996).

Como as interações entre os agentes são guiadas pelas informações que eles mantém uns sobre os outros, durante estas interações eles podem detectar se tais informações são incorretas ou incompletas, e eventualmente decidir revisá-las para manter a consistência. Esta característica é extremamente importante no contexto de SMA abertos, onde certamente uma situação onde os agentes tenham

um conhecimento completo e correto uns dos outros não é uma a regra, mas sim a exceção.

8.2.2 Princípios sobre o comportamento dos agentes

No sistema $\mathcal{D}_{\text{EPINT}}$, foram adotados 4 (quatro) princípios fundamentais sobre o comportamento dos agentes:

- 1. Princípio da Não Benevolência: os agentes decidem de forma autônoma se desejam ou não cooperar com os outros, não existindo uma imposição de assim fazê-lo a priori;
- 2. Princípio da Sinceridade: os agentes nunca decidem informar fatos incorretos aos outros de modo deliberado, sendo que apenas comunicam os fatos nos quais acreditam;
- 3. Princípio do Auto-Conhecimento: os agentes têm um conhecimento correto e completo sobre si próprios, enquanto que suas crenças relativas aos outros agentes podem ser incorretas, devido a falhas em suas fontes de informação;
- 4. Princípio da Consistência: quando detectam uma inconsistência nas informações que mantêm uns sobre os outros, os agentes procuram eliminá-la, restaurando um estado de informação consistente.

8.2.3 Arquitetura dos agentes

A arquitetura adotada para os agentes do sistema \mathcal{D} EPINT é a arquitetura ASIIC (BOISSIER, 1993; BOISSIER; DEMAZEAU, 1994), apresentada na figura 8.2.

De forma concisa, nesta arquitetura um agente é composto por vários *mecanismos internos*, um dos quais é o mecanismo de raciocínio social, que utiliza as informações armazenadas na descrição externa, descrita na seção 5.1.4.

Nos agentes do sistema $\mathcal{D}_{\text{EPINT}}$, foram adotadas ainda as seguintes simplificações:

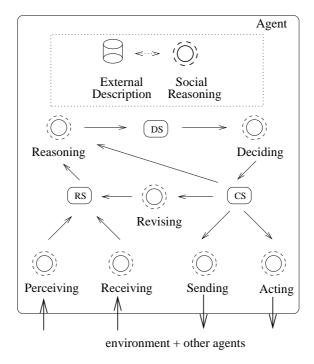


Figura 8.2: Arquitetura de agente ASIC (SICHMAN, 1995).

- os agentes não realizam um procedimento de planejamento "on-line", e caso não possuam um plano para atingir um de seus objetivos adotam incondicionalmente o plano de um eventual proponente de uma proposta de coalizão;
- 2. na ativação de seus mecanismos internos, os agentes escolhem respectivamente nesta ordem (i) um objetivo a atingir; (ii) um plano cuja execução atinja o objetivo escolhido e (iii) parceiros a quem propostas de coalizão são enviadas, no caso do agente ser dependente dos outros para este objetivo, conforme a técnica de CBD, discutida na seção 5.1.11;
- 3. os mecanismos internos de percepção e inferência são simulados por uma interface com o usuário;
- 4. como discutido na seção 5.1.11, o sistema se limita a tratar OU-dependências, sendo consideradas apenas coalizões envolvendo 2 (dois) agentes¹;
- 5. as propostas de coalizão são tratadas de modo sequencial, não sendo permitido um tratamento concorrente de mais de uma proposta.

 $^{^{1}}$ A situação onde existem E-dependências, cujo modelo é discutido na seção 5.1.12, é implementada no sistema \mathcal{D} EPINT $^{+}$, descrito na seção 8.3.

8.2.4 Funcionamento do sistema

Devido à ausência do tratamento de concorrência de propostas de coalizão, foram criados 2 (dois) comportamentos básicos para os agentes do sistema:

- 1. comportamento ativo, onde o agente executa os passos da técnica CBD descrita na seção 5.1.11, ao utilizar seu mecanismo de raciocínio social para escolher um objetivo, um plano e um possível parceiro a quem envia uma proposta de coalizão, e aguarda uma resposta deste parceiro, ativando posteriormente o procedimento de revisão de crenças;
- 2. comportamento passivo, onde ocorrem os seguintes passos:
 - (a) inicialmente, o usuário simula a ativação dos mecanismos de percepção e raciocínio do agente, atualizando a sua descrição externa;
 - (b) o agente trata então as mensagens que lhe foram enviadas, na ordem de chegada, que podem estar relacionadas à entrada ou saída de um agente da sociedade (quando então o agente atualiza a sua descrição externa) ou então a uma proposta de coalizão, à qual o agente responde após um processo interno de decisão;
 - (c) o agente ativa seu procedimento de revisão de crenças.

Cabe ao usuário do sistema determinar qual comportamento será ativado em cada agente do sistema, sendo que apenas um agente pode ter o comportamento ativo num determinado ciclo.

As respostas possíveis a uma proposta de formação de coalizão são as seguintes (SICHMAN, 1995):

- 1. aceitação, sendo que neste caso a coalizão é formada e o sistema retorna ao estado inicial, possibilitando o início de um novo ciclo de formação de coalizões;
- 2. recusa, sendo que neste caso o agente ativo tenta buscar, sucessivamente, novos parceiros, planos ou objetivos a atingir, segundo a técnica CBD descrita na seção 5.1.11;

3. revisão, sendo que neste caso o agente passivo informa ao agente ativo que este tem uma informação incorreta ou incompleta sobre ele, e sob a qual foi baseada a proposta de coalizão.

Tanto o agente ativo (ao receber uma resposta de revisão), quanto o agente passivo (ao receber uma proposta e perceber, por exemplo, que sua informação sobre o agente ativo é incompleta) ativam um procedimento de revisão de crenças, cujos detalhes são descritos em (SICHMAN, 1995; SICHMAN; DEMAZEAU, 1995a; SICHMAN; DEMAZEAU, 1996).

Ao receber uma mensagem de revisão, o agente ativo busca novos parceiros, planos ou objetivos a atingir, de maneira semelhante ao caso da mensagem de recusa.

Os agentes ativo e passivo utilizam os critérios de escolha e aceitação de parceiros baseados nas situações de objetivo e de dependência da técnica CBD, apresentados na seção 5.1.11.

8.2.5 Protocolo de formação de coalizão

O sistema \mathcal{D} EPINT utiliza a noção de protocolo de comunicação, descrita na seção 3.6.4, para estruturar o processo de formação de coalizão entre agentes.

Foram desenvolvidos 3 (três) protocolos para este sistema:

- 1. protocolo de apresentação, utilizado pelos agentes ao entrar no sistema, quando enviam aos outros as informações sobre seus objetivos, planos, ações e recursos;
- 2. protocolo de saída, utilizado pelos agentes que saem do sistema para que os restantes possam atualizar as suas descrições externas, e particularmente atualizar a lista de ações disponíveis no sistema;
- 3. protocolo de formação de coalizões, mostrado na figura 8.3, que estrutura a troca de de mensagens adequadas para implementar a técnica CBD, apresentada na seção 5.1.11.



- 1.(all)(request, bargaining, dec, depintproposition, receive)(proposition, data)
- 2.(if error)(you)(answer, warning, obs, depintproposition, end)(revision, data)
- 3.(if (!error and !best_option))(you)(answer,informing,dec,depintproposition,end)(refusal)
- 4.(if (!error and best_option))(you)(answer,informing,dec,depintproposition,end)(acceptance)

Figura 8.3: Protocolo de comunicação do sistema DEPINT (SICHMAN, 1995).

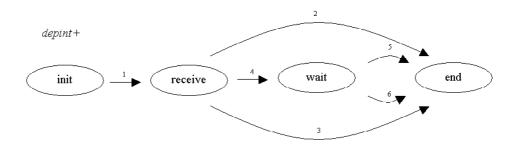
O sistema \mathcal{D} epint é executado em plataformas UNIX, e foi desenvolvido na linguagem C++. Utiliza a plataforma \mathcal{M} ASENV, descrita na seção 7.1, para implementar a comunicação entre agentes, inclusive o protocolo mostrado na figura 8.3. No apêndice III, exibem-se os resultados de uma sessão de utilização do sistema \mathcal{D} epint, que ilustram um exemplo de formação de uma coalizão entre agentes, num domínio de laboratórios de pesquisa acadêmicos.

8.3 Aplicação \mathcal{D} epint⁺

 \mathcal{D} EPINT⁺ (DAVID, 1998; DAVID; SICHMAN; COELHO, 1999; DAVID; SICHMAN; COELHO, 2000) é um sistema que implementa a técnica de coalizões baseadas em dependência extendidas, discutida na seção 5.1.12. Sua estrutura interna é muito semelhante à do sistema \mathcal{D} EPINT, descrito na seção anterior, com as seguintes diferenças:

- o sistema tem uma interface homem-máquina mais agradável, baseada em interfaces gráficas;
- 2. o critério de escolha e aceitação de parceiros leva em conta agora os eventuais custos de coordenação, conforme discutido na seção 5.1.12;
- 3. o protocolo de formação de coalizões, mostrado na figura 8.4, tem um estado interno adicional (wait) para poder tratar de modo conveniente as situações de E-dependência, conforme também discutido na seção 5.1.12.

O sistema \mathcal{D} EPINT⁺ é executado em plataformas UNIX, e foi desenvolvido na linguagem C++. Também utiliza a plataforma \mathcal{M} ASENV, descrita na seção 7.1,



- 1.(all)(request,bargaining,dec,depintproposition,receive)(proposition,data)
- 2.(if error)(you)(answer,warning,obs,depintproposition,end)(revision,data)
- 3.(if (!error and !best_option))(you)(answer,informing,dec,depintproposition,end)(refusal)
- 4.(if (!error and best_option))(you)(answer,informing,dec,depintproposition,wait)(acceptance)
- $5. (if (suc_coalition) (you) (answer, confirming, dec, depint proposition, end) (confirmation) \\$
- 6.(if (!suc coaltion)(you)(answer,cancelling,dec,depintproposition,end)(cancelation)

Figura 8.4: Protocolo de comunicação do sistema $\mathcal{D}_{\text{EPINT}^+}$ (DAVID, 1998).

para implementar a comunicação entre agentes, inclusive o protocolo mostrado na figura 8.4. A figura 8.5 mostra a interface principal do sistema.

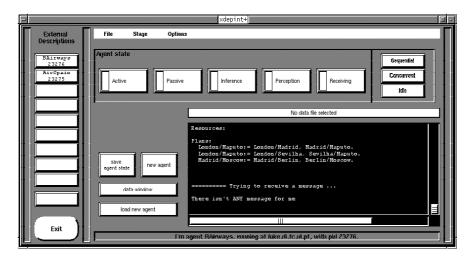


Figura 8.5: Interface do sistema \mathcal{D} EPINT⁺ (DAVID, 1998).

No apêndice IV, exibem-se os resultados de uma sessão de utilização do sistema $\mathcal{D}_{\text{EPINT}^+}$, que ilustram um exemplo de tentativa de formação de uma coalizão entre agentes, num domínio de aviação comercial.

 $8.4 \quad Conclus\~oes$ 151

8.4 Conclusões

A figura 8.6 mostra as aplicações em raciocínio social classificadas segundo os eixos básicos propostos na seção 2.3.

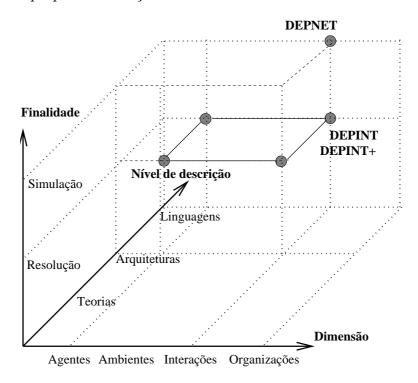


Figura 8.6: Classificação das aplicações de raciocínio social.

Em relação à sua finalidade, enquanto o sistema \mathcal{D} EPNET tem uma finalidade de simulação social, os sistemas \mathcal{D} EPINT e \mathcal{D} EPINT⁺ adotam uma perspectiva de resolução social. O sistema \mathcal{D} EPNET se situa no nível de descrição de linguagens e aborda exclusivamente a dimensão de interação entre agentes da metodologia Vogais. Já os sistemas \mathcal{D} EPINT e \mathcal{D} EPINT⁺ utilizam 2 (dois) níveis de descrição: arquitetura (para os agentes que os constituem) e linguagens. Quanto às dimensões da metodologia Vogais, ambos os sistemas abordam tanto a dimensão referente aos agentes quanto a dimensão referente às suas interações.

Finalmente, no que diz respeito à representação das interações entre agentes, discutida na seção 3.3, enquanto nos sistemas \mathcal{D} EPINT e \mathcal{D} EPINT⁺ o modelo de interação adotado é intencional, no sistema \mathcal{D} EPNET considera-se um modelo não intencional, pois um observador externo é responsável por identificar as possíveis interações entre os agentes.

9 Aplicações em Raciocínio Organizacional

Neste capítulo, são apresentadas respectivamente nas seções 9.1, 9.2, 9.3 e 9.4 as aplicações RAISORG, COLLORG, JOJTEAM e COMMUNET. Tais aplicações são descritas de forma resumida, de modo a realçar suas características principais. Na seção 9.5, estas aplicações são situadas em relação à classificação proposta na seção 2.3.

Mais detalhes sobre as aplicações \mathcal{R} AISORG, \mathcal{C} OLLORG e \mathcal{J} OJTEAM podem ser encontrados respectivamente em (HANNOUN et al., 1998; HANNOUN, 2002), (HANNOUN, 2002) e (HÜBNER, 2003).

9.1 Aplicação \mathcal{R} aisorg

RAISORG (HANNOUN et al., 1998; HANNOUN, 2002) é uma ferramenta de auxílio à concepção e de simulação de funcionamento de uma organização baseada no modelo Moise, descrito na seção 6.1. A ferramenta, desenvolvida em ambiente UNIX na linguagem Java, implementa o cálculo das dependências e coerências organizacionais, discutidas nas seções 6.1.6 e 6.1.7. As seguintes funcionalidades estão presentes na ferramenta:

- construção e exibição da árvore de dependências entre missões (nível estrutura organizacional);
- construção e exibição da árvore de dependências entre agentes (nível entidade organizacional);
- verificação e exibição da coerência da estrutura organizacional;

- verificação e exibição da coerência da entidade organizacional;
- aplicação de heurísticas simples para solucionar os problemas associados a eventuais incoerências.

A título de ilustração de funcionamento da ferramenta, seja um exemplo referente ao cenário do programa de pós-graduação, cuja especificação organizacional é representada na tabela 9.1.

Estrutura Organizacional							
Papel	Missão	Objetivo	Plano	Ação	Recurso		
Professor (ro_1)	(\mathbf{P},m_1)	$\{g_1\}$	$\{p_1(g_1)=a_1;\ a_2(r_1);\ a_3.\}$	$\{a_1\}$	Ø		
	(\mathbf{O},m_2)	$\{g_2\}$	$\{p_2(g_2)=a_4;\ a_5.\}$	$\{a_3\}$	Ø		
Secretária (ro ₂)	(\mathbf{P}, m_3)	Ø	Ø	$\{a_2, a_3\}$	$\{r_1,r_2\}$		
Aluno (ro_3)	(\mathbf{O},m_4)	Any_{g}	Any_{p}	$\{a_5\}$	Ø		
Relações de comunicação							
$\text{entre} \ll Professor \gg \text{e} \ll Secret\'{a}ria \gg : \ l_1 = \langle ro_1, ro_2, communication, \{m_1\}, \{m_3\}, \{pr_1\} \rangle$							
$\text{entre} \ll Professor \gg \text{e} \ll Aluno \gg \ : \ l_2 = \langle ro_1, ro_3, communication, \{m_2\}, \{m_4\}, Any_{pr} \rangle$							
Relações de Autoridade							
$\text{entre} \ll Professor \gg \text{e} \ll Aluno \gg : \ l_3 = \langle ro_1, ro_3, authority, \{m_2\}, \{m_4\}, \{l_2\} \rangle$							
Relações de Conhecimento							
$\text{entre} \ll Secret \acute{a}ria \gg \text{e} \ll Aluno \gg : \ l_4 = \langle ro_2, ro_3, acquaintance, \{m_3\}, \{m_4\}, \{l_2, l_3\} \rangle$							

g1: EscreverApostila; g2: MinistrarCurso. a1: ConceberApostila; a2: Datilografar; a3: Imprimir; a4: MinistrarAulas; a5: AssistirAulas. r1: LaTeX; r2: Wordperfect.

Tabela 9.1: Exemplo de especificação organizacional para aplicação \mathcal{R} AISORG (HANNOUN, 2002).

Neste cenário, existem 3 (três) papéis, « Professor », « Secretária » e « Aluno », bem como 2 (duas) relações de comunicação e 1 (uma) única relação de autoridade. O papel de « Professor » interage com o papel de « Secretária » para atingir o objetivo « EscreverApostila », e também interage com o papel de « Aluno » no escopo do objetivo « MinistrarCurso ».

Já a entidade organizacional encontra-se representada na tabela 9.2. Definem-se 4 (quatro) agentes, que desempenham os papéis da organização, e cada um deles é capaz de executar algumas ações e possui certos recursos.

Entidade Organizacional					
Agente	Papel	Competência			
Agente		Ações	Recursos		
agent1	Professor	a1, a3, a4	r1		
agent2	Secretária	a2	r1, r2		
agent3	Aluno	a5			
agent4	Aluno	a5			

 $a1: Conceber Apostila\,;\,\, a2: \, Datilografar\,;\,\, a3: \, Imprimir\,; \\ a4: \, Ministrar Aulas\,;\,\, a5: \, Assistir Aulas. \\ r1: \, \LaTeX\,;\,\, r2: \, Wordperfect.$

Tabela 9.2: Exemplo de entidade organizacional para aplicação \mathcal{R} AISORG (HANNOUN, 2002).

9.1.1 Cálculo de dependências organizacionais

O cálculo de dependências organizacionais é realizado em dois níveis distintos, referentes à estrutura e à entidade organizacionais, conforme discutido na seção 6.1.6.

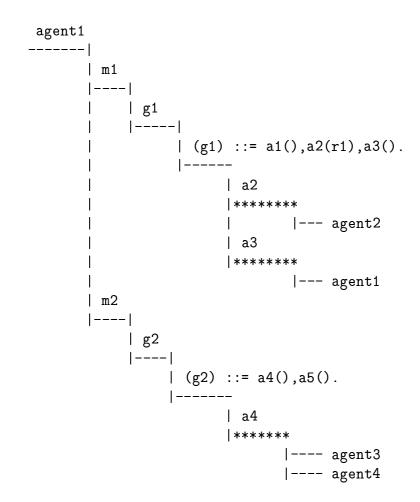
9.1.1.1 Dependências na estrutura organizacional

Seja a missão « m1 » associada ao papel de « Professor ». A ferramenta constrói a seguinte árvore de dependência entre missões:

Esta árvore demonstra que as missões « m1 » e « m3 » estão relacionadas entre si.

9.1.1.2 Dependências na entidade organizacional

Considere-se agora o agente agent1. A ferramenta constrói a seguinte $\acute{a}rvore\ de$ $depend\hat{e}ncia\ entre\ agentes$:



9.1.2 Detecção de incoerências organizacionais

Analogamente ao caso das dependências organizacionais, a validação da coerência organizacional é realizada em dois níveis distintos, referentes à estrutura e à entidade organizacionais, conforme discutido na seção 6.1.7.

9.1.2.1 Incoerências na estrutura organizacional

O cálculo de coerência da estrutura organizacional é baseado nas noções de *autonomia* e de *coerência* de um papel.

Por exemplo, a supressão da ação « a_1 » do conjunto de ações permitidas no

contexto da missão « m_1 » gera uma incoerência organizacional:

Le role "Professeur" n'est pas autonome dans la SO

- => aucune mission de la SO n'autorise l'action a1.
- => l'action instanciee a1() du plan p1 de la mission m1 est non realisable.

Seguindo o mesmo princípio, se a relação de comunicação que liga a missão $\ll m_1 \gg$ do papel $\ll Professor \gg$ à missão $\ll m_3 \gg$ do papel $\ll Secretária \gg$ for suprimida, também ocorre um diagnóstico de uma incoerência na estrutura organizacional:

Le role ''Professeur'' n'est pas coherent dans la SO

- => la mission m1 depend de la mission m3 pour l'action instanciee a2(r1).
 - * aucun lien ne relie la mission m1 a la mission m2.
- => la mission m1 depend de la mission m3 pour l'action instanciee a3().
 - * aucun lien ne relie la mission m1 a la mission m2.

9.1.2.2 Incoerências na entidade organizacional

O cálculo de coerência da entidade organizacional é baseado nas noções de autonomia de um agente e de coerência de atribuição de papéis.

A título de exemplo, a supressão da ação « a_1 » do conjunto de ações do agente agent1 gera uma incoerência detectada pela aplicação da seguinte maneira:

L'agent 'agent1' n'est pas autonome dans l'EO

- => aucun agent ne dispose de l'action a1.
- => l'action instanciee a1() du plan p1 de la mission m1 du role Professeur de l'agent ''agent1'' est non realisable.

No que diz respeito à coerência de atribuição de papéis, se a ação $\ll a_4 \gg$ for suprimida do conjunto de ações do agente agent1 e for adicionado ao conjunto de ações do agente agent4, a seguinte incoerência é detectada pela aplicação:

Incoherence dans la distribution des roles

- => toutes les actions necessaires au plan p2 de m2 sont disponibles.
 - * l'action instanciee a4() est necessaire au plan p2.
 - * l'agent agent4 dispose de l'action a4.
 - * l'agent agent4 ne peut executer l'action instanciee a4() dans le cadre de son role.

9.1.3 HEURÍSTICAS PARA RESTABELECER A COERÊNCIA ORGANIZACIONAL

Como discutido na seção 6.1.6, um possível contexto de aplicação do modelo \mathcal{M} OISE é o domínio de avaliação organizacional. Nas seções anteriores deste capítulo, mostrou-se como a ferramenta \mathcal{R} AISORG é capaz de detectar algumas incoerências organizacionais.

Uma vez detectadas as incoerências no nível da estrutura e da entidade organizacional, algumas modificações simples podem ser sugeridas para restaurar o bom funcionamento do sistema. A ferramenta \mathcal{R} AISORG sugere ao usuário uma modificação automática, quando possível, que possa restaurar a coerência organizacional. As heurísticas utilizadas para este fim são apresentadas na tabela 9.3.

Retomando o exemplo apresentado na seção 9.1.2, quando foi suprimida a relação de comunicação entre os papéis de « Professor » e de « Secretária », a ferramenta \mathcal{R} AISORG propõe a seguinte heurística para restaurar a coerência da entidade organizacional:

Le role ''Professeur'' n'est pas coherent dans la SO

- => la mission m1 depend de la mission m3 pour l'action instanciee a2(r1).
 - * aucun lien ne relie la mission m1 a la mission m2.
- => la mission m1 depend de la mission m3 pour l'action instanciee a3().
 - * aucun lien ne relie la mission m1 a la mission m2.

voulez-vous une corriger ce probleme (o/n) ? o
proposition(s):

Estrutura organizacional	
Incoerência	Solução
papel não autônomo	acrescentar o objetivo, plano ou ação, com
	os recursos associados, à missão que causa o
	fenômeno
papel não coerente	criar uma relação de comunicação entre a
	missão do papel que causa o fenômeno e a
	missão da qual ela depende e cuja relação
	está ausente
Entidade organizacional	
Incoerência	Solução
agente não autônomo	sem solução, nenhum agente dispõe da com-
	petência necessária
atribuição de papéis não coerente	atribuir ao agente do qual se depende para a
	execução da ação instanciada um papel que
	torne a delegação possível

Tabela 9.3: Heurísticas para restaurar a coerência organizacional (HANNOUN, 2002).

- (1) autoriser l'action a2, l'action a2 et la ressource r1 pour la mission m1.
- (2) ajouter un lien de communication entre la mission m1 et la mission m2.

introduire un choix : 2
voulez-vous sauvegarder la nouvelle SO issue de ce changement (o/n) ? o
introduire un nom pour le fichier de la SO : exemple.so
le fichier a ete sauvegarde sous le nom ''exemple.so''

9.2 Aplicação \mathcal{C} ollorg

Collorg (Hannoun, 2002) é um sistema que ilustra a utilização do modelo Moise no domínio de "groupware". Como exemplo concreto, utilizou-se um cenário simplificado de alguns procedimentos administrativos referentes ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Escola Politécnica

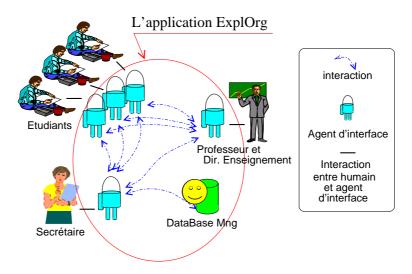


Figura 9.1: Arquitetura do sistema Collorg (HANNOUN, 2002)

da Universidade de São Paulo (USP)¹.

O sistema se propõe a oferecer 2 (duas) funcionalidades²:

- 1. oferecer um serviço "on-line" para informar aos ingressantes e demais interessados as regras e os procedimentos administrativos a seguir, bem como as pessoas a contactar;
- 2. oferecer serviços de base para o trabalho cooperativo, de modo a automatizar, quando possível, algumas tarefas administrativas repetitivas.

A aplicação, desenvolvida em ambiente UNIX na linguagem Java, utiliza a plataforma \mathcal{M} AST (BOISSIER et al., 1998) para implementar a comunicação entre agentes.

A arquitetura do sistema, representada na figura 9.1, utiliza 2 (dois) tipos de agentes: os agentes de interface, associados aos usuários, e os agentes de software, como por exemplo o gestor da base de cursos. Os agentes de interface são implementados como "applets", acessíveis via uma Intranet.

A janela principal do agente de interface é apresentada na figura 9.2. Esta janela é dividida em 3 (três) regiões:

¹Tais procedimentos eram válidos em 1997/1998, quando o autor desta tese era o coordenador responsável pela área de concentração em sistemas digitais do referido programa.

²Neste manuscrito, a apresentação do sistema será limitada à primeira funcionalidade. Uma descrição da segunda pode ser encontrada em (HANNOUN, 2002).

- a região de conexão (na parte superior da tela), que permite que o usuário se identifique e se conecte;
- a região de exploração (na parte central da tela), que permite ao usuário visualizar a definição de suas missões, bem como identificar os interlocutores dos quais depende;
- a região de ajuda (na parte inferior da tela), que exibe um texto explicativo sobre o último elemento selecionado pelo usuário.

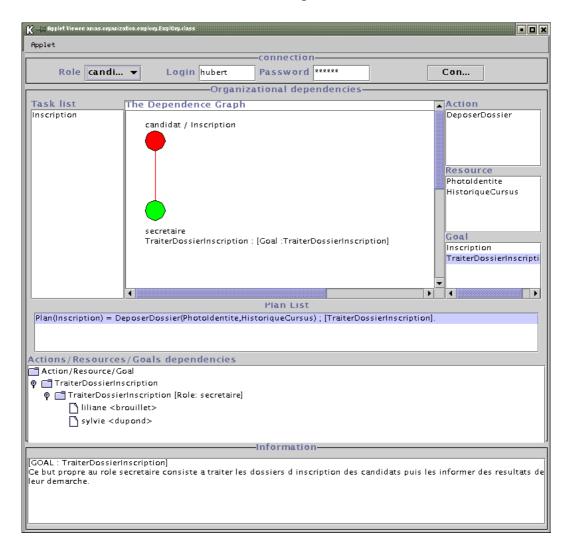


Figura 9.2: Agente de interface do sistema COLLORG (HANNOUN, 2002).

O usuário pode também selecionar na região de conexão a descrição de outros papéis que eventualmente deseje conhecer, desde que esteja ligado a eles por relações organizacionais, como descritas na seção 6.1.3. Entretanto, as missões que serão exibidas se limitam àquelas ligadas pela relação organizacional em questão.

A região de exploração permite visualizar as missões, bem como os objetivos, planos, ações e recursos a ela associados. Exibe também as relações de dependência entre os papéis, conforme descrito na seção 6.1.6.

No exemplo representado na figura 9.2, o usuário está desempenhando o papel de « Candidat » e só dispõe da missão « Inscription », que depende da missão « TraiterDossierInscription », associada ao papel de « Secretaire », como indicado no grafo de dependência da estrutura organizacional, exibido no centro da janela. Logo abaixo, o plano indica o procedimento a ser seguido.

Um pouco mais abaixo, a árvore de dependência, relativa à entidade organizacional, indica que para executar a missão « Inscription », o candidato depende de duas pessoas: Liliane BROUILLET e Sylvie DUPOND. Tais pessoas desempenham o papel de « Secretaire », e são as únicas capazes de ativar o sub-objetivo « Traiter Dossier Inscription » definido no plano. A região de ajuda exibe uma mensagem explicativa sobre o objetivo « Traiter Dossier Inscription », selecionado na lista de objetivos.

9.3 Aplicação \mathcal{J} ojteam

Jojteam (HÜBNER, 2003) é uma aplicação em desenvolvimento que utiliza o modelo Moise⁺, em particular o processo de reorganização apresentado na seção 6.2.6, no domínio de futebol de robôs simulados (KITANO et al., 1995), que é ilustrado na figura 9.3 através do simulador TeamBots. Este domínio apresenta a maioria das propriedades de um problema adequado ao uso de SMA: autonomia e reatividade dos agentes, distribuição, controle descentralizado e necessidade de coordenação. Além destas propriedades, dado o caráter dinâmico do ambiente, o time precisa se adaptar às mudanças do time oponente, tornando-se, portanto, um bom candidato a ambiente de testes para os modelos de reorganização (STONE; VELOSO, 1999).

A reorganização do \mathcal{J} OJTEAM é realizada por um grupo destinado a esta finalidade, denominado « ReorgGr », cuja especificação organizacional é apresentada na seção 6.2.6. Os agentes que participam deste grupo não são os jogadores, mas sim agentes especializados nas tarefas de monitorar, projetar e selecionar organizações.



Figura 9.3: Tela do simulador $\mathcal{T}_{EAMBOTS}$ (TEAMBOTS, 2000).

As etapas do processo de reorganização, discutidas na seção 4.9, são realizadas pelos agentes do \mathcal{J} OJTEAM da seguinte maneira:

Monitoração do time O monitoramento no Jojteam é feito de uma forma singular. No domínio de futebol de robôs simulado, o agente de monitoração inicia uma reorganização a cada 24.000 passos de simulação. Como o jogo tem 120.000 passos, existem 5 processos de reorganização em cada jogo, independentemente de seu placar corrente. A primeira reorganização acontece no passo 0, a segunda no passo 24.000, e a última no passo 96.000. Como a reorganização sempre acontece, mesmo que o time esteja jogando bem, tal procedimento permite a exploração de novas, e eventualmente melhores, organizações.

Projeto de organizações Os projetos de reorganização são enviados por vários agentes no domíno. Cada um deles pode ter uma estratégia distinta, como por exemplo ser mais arrojado e colocar o time para frente, ou ser mais cauteloso para garantir um resultado. Alguns exemplos de propostas seriam os seguintes:

- alterar a formação do time para 1x1x3 (1 goleiro, 1 zagueiro e 3 atacantes),
- alterar a formação para 1x3x1,
- alterar a formação para 4x1,
- mudar a área de atuação dos jogadores no campo,
- mudar as metas do jogadores, etc.

Seleção de projetos O agente de seleção, em desenvolvimento, poderá utilizar técnicas de aprendizado por reforço (SUTTON; BARTO, 1998) para aprender a identificar qual a melhor proposta para um determinada situação do jogo. Este agente poderia, por exemplo, aprender que no início do jogo é melhor escutar a proposta do projetista mais arrojado (1x1x3), e se perto do final do jogo seu time estiver ganhando, selecionar então a proposta do projetista mais cauteloso (4x1).

9.4 Aplicação \mathcal{C} ommunet

A busca de informações na Internet é atualmente um tema de destaque na pesquisa em computação. O uso de máquinas de busca convencionais para auxiliar nesta tarefa fornece uma solução geral. No entanto, este enfoque apresenta alguns problemas, tais como (i) mapeamento incompleto do conteúdo acessível na rede; (ii) algoritmos de categorização ("ranking") satisfatórios para casos gerais, mas com utilização insatisfatória para casos particulares e (iii) categorizações que podem estar viciadas por motivos econômicos, políticos, etc.

Uma alternativa interessante consiste na utilização de recursos presentes na rede de contatos sociais, formada pelas pessoas com as quais um usuário interage. A exploração computacional do chamado fenômeno do mundo pequeno³ pode tornar possível o acesso a páginas, documentos e pessoas cuja existência não está necessariamente mapeada pelas máquinas de busca convencionais.

O sistema COMMUNET, em desenvolvimento, foi concebido com base em tal idéia. Baseado em contatos entre pares, fornece uma alternativa decentralizada

³O número mínimo de pessoas contatadas para atingir um membro de uma comunidade humana é tipicamente muito pequeno, não chegando a uma dezena em média (TRAVERS; MILGRAM, 1969).

que complementa os enfoques convencionais de busca de informações, oferecendo uma liberdade maior ao usuário e repousando em algo familiar para as pessoas: a comunidade na qual elas convivem.

O elemento principal do sistema é o chamado agente mediador, capaz de representar e raciocinar socialmente sobre os contatos de seu usuário, no contexto de um SMA aberto, conforme discutido na seção 2.2.2.

Cada usuário tem um agente mediador associado, que obtém informações sobre seus recursos, sua visão subjetiva da rede social em que está inserido (dada pelas trocas de e-mail) e suas permissões de acesso relacionadas aos seus contatos. O usuário pode também associar grupos e eventualmente papéis aos seus contatos, conforme discutido nos modelos apresentados no capítulo 6, estabelecendo assim a sua visão subjetiva da organização na qual está inserido. Estes dados, obtidos pelo agente mediador de forma semi-automática, permitem com que ele gerencie a interação com agentes mediadores de outros contatos, incluindo uma troca de recursos associados aos usuários que representam. Os recursos podem ser disponibilizados diferentemente pelos usuários, conforme uma política própria de privacidade.

A interface do agente mediador do sistema \mathcal{C} OMMUNET é mostrada na figura 9.4, e contém 3 (três) áreas distintas: (i) uma área para receber a consulta do usuário (área 1); (ii) outra área para exibir os recursos relacionados à consulta obtidos de outros contatos (área 2) e (iii) uma terceira área para apresentar os resultados de consultas às máquinas de busca convencionais (área 4).

A título de exemplo, seja um usuário fictício chamado Carlos Dias, que pretende realizar uma busca de informações sobre a linguagem Prolog. Uma possível seqüência de interação entre o usuário e seu agente mediador agCARLOS seria a seguinte:

- 1. o usuário digita a consulta "prolog" na área 1;
- 2. não conseguindo localizar a palavra na sua representação do conhecimento do usuário, o agente agCARLOS encaminha esta palavra a outros agentes da comunidade, sendo que este passo poderia incorporar a apresentação ao usuário de um dicionário (por exemplo, Wordnet) ou de uma árvore de temas genéricos (por exemplo, Yahoo) para tentar diminuir a ambigüidade

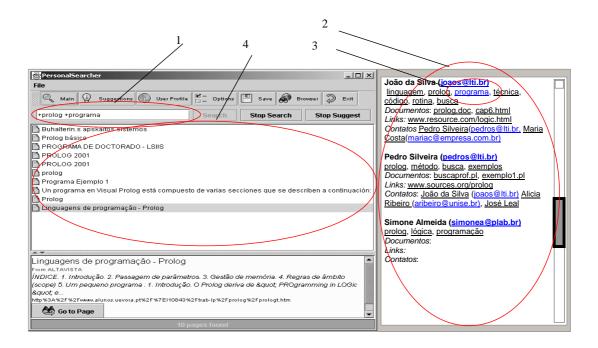


Figura 9.4: Interface de um agente mediador no sistema \mathcal{C} OMMUNET.

da requisição;

- 3. o agente agCARLOS apresenta, na área 2, os nomes dos contatos cujos agentes mediadores responderam à requisição, juntamente com os recursos que cada um deles disponibilizou na forma de palavras, páginas, documentos e contatos pessoais relacionados ao(s) tema(s);
- 4. o usuário decide adicionar a palavra "programa", destacada na área 3 da figura 9.4 como tendo sido recebida do agente $agJO\tilde{A}O$, associado ao usuário João da Silva, à sua consulta;
- 5. uma busca na Internet é ativada, também intermediada pelo agente ag-CARLOS, e seu resultado exibido ao usuário na área 4;
- 6. o acesso do usuário às páginas sugeridas é monitorado pelo agente agCAR-LOS, que ao término da sessão de utilização envia aos agentes mediadores que forneceram recursos (no exemplo, o agente $agJO\tilde{A}O$) um retorno consistente do grau de utilidade/uso de tais recursos, como por exemplo qual recurso foi utilizado, quais páginas foram efetivamente lidas, etc.

O agente mediador do sistema $\mathcal{C}_{\text{OMMUNET}}$ é baseado no agente de busca PersonalSearcher, descrito em (LUGO et al., 2002).

3

 $9.5 \quad Conclus\~oes$

9.5 Conclusões

A figura 9.5 mostra as aplicações em raciocínio organizacional classificadas segundo os eixos básicos propostos na seção 2.3.

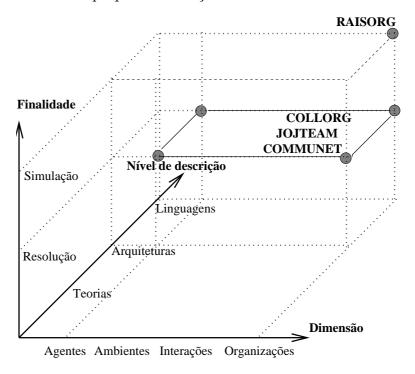


Figura 9.5: Classificação das aplicações de raciocínio organizacional.

Em relação à sua finalidade, enquanto o sistema RAISORG tem uma finalidade de simulação social, os sistemas COLLORG, JOJTEAM e COMMUNET adotam uma perspectiva de resolução social. O sistema RAISORG se situa no nível de descrição de linguagens e aborda exclusivamente a dimensão de organização entre agentes da metodologia Vogais. Já os sistemas COLLORG, JOJTEAM e COMMUNET utilizam 2 (dois) níveis de descrição: arquitetura (para os agentes que os constituem) e linguagens. Quanto às dimensões da metodologia Vogais, estes 3 (três) sistemas abordam tanto a dimensão referente aos agentes quanto a dimensão referente às suas organizações.

Finalmente, no que diz respeito à representação das organizações de agentes, discutida na seção 4.8, enquanto no sistema \mathcal{R} AISORG o modelo de interação adotado é não intencional, pois cabe a um observador externo identificar a organização dos agentes, nos sistemas \mathcal{C} OLLORG, \mathcal{J} OJTEAM e \mathcal{C} OMMUNET considera-se um modelo intencional de representação destas organizações.

10 Outras Aplicações Multiagentes

Neste capítulo, são apresentadas respectivamente nas seções 10.1, 10.2 e 10.3 as aplicações multiagentes \mathcal{P} artnet⁺, \mathcal{O} ptima e 2 (duas) implementações desenvolvidas no projeto \mathcal{N} alamas, que embora não utilizem diretamente os modelos sociais e organizacionais apresentados na parte II, ilustram outros usos de conceitos relativos à interação e organização de agentes apresentados na parte I. Na seção 10.4, estas aplicações são situadas em relação à classificação proposta na seção 2.3.

Mais detalhes sobre as aplicações \mathcal{O} PTIMA e sobre aquelas desenvolvidas no projeto \mathcal{N} ALAMAS podem ser encontrados respectivamente em (RABAK, 1999; RABAK; SICHMAN, 2001) e (PAIVA, 1997; LIMA et al., 1997; SILVA; ABRAHÃO; LIMA, 1998; CARVALHO et al., 1998).

10.1 Aplicação \mathcal{P} artnet⁺

O simulador \mathcal{P} ARTNET⁺, em fase de desenvolvimento, propõe algumas extensões a um simulador denominado \mathcal{P} ARTNET (CONTE; PEDONE, 1998). Este simulador busca comprovar, através da técnica de simulação baseada em multiagentes, uma teoria social sobre a importância da existência de agentes com diferentes perfis para o estabelecimento de parcerias numa sociedade. Tal teoria visa reforçar a distinção entre duas visões a respeito da racionalidade: a visão substancialista, onde os objetivos finais tem relevância, e a visão instrumentalista, muito em voga nos dias de hoje, onde os objetivos finais não tem importância, e onde os agentes se limitam a procurar atingir o ponto de equilíbrio entre ganho e custo. Esta segunda visão, baseada na Teoria dos Jogos (AXELROD, 1984), gera incertezas

quando há múltiplos pontos de equilíbrio.

Os agentes do simulador têm objetivos a atingir, e são capazes de desempenhar ações. Não existe, entretanto, a noção de plano, como descrita na seção 5.1.4, pois cada ação é suficiente para atingir um objetivo. Deste modo, as parcerias são estabelecidas sempre entre apenas 2 (dois) agentes. Além disso, os agentes nunca são autônomos para atingir seus objetivos, e procuram parceiros na sociedade para ajudá-los a atingir tais objetivos. Conseqüentemente, como descrito na seção 5.1.8, são estabelecidas relações de dependência recíproca entre os agentes, que eventualmente os conduzem a um escambo social. Cada ação tem um custo associado e cada objetivo tem uma importância associada. Dependendo da estratégia adotada por cada agente, são escolhidos os parceiros mais adequados.

Existem 3 (três) tipos básicos de agentes no simulador \mathcal{P} ARTNET, que diferem em relação à estratégia de escolha de parceiros:

- 1. *utilitaristas* (ou instrumentalistas), que tentam maximizar a diferença entre o benefício (importância dos objetivos atingidos) e os custos (associados às ações realizadas pelos parceiros);
- 2. substancialistas (ou hedonistas), que não se importam com os custos, mas sim com a ordem de prioridade dos seus objetivos;
- 3. avaros (ou miseráveis) que ordenam as alternativas de parceria de forma a minimizar os seus custos, independentemente da importância de seus objetivos.

A simulação se passa em ciclos, onde em cada passo um agente, escolhido aleatoriamente, tem a possibilidade de escolher um objetivo e um parceiro para atingí-lo. Se um agente for escolhido como parceiro por mais de um agente num ciclo, ele tem o direito de escolher com qual deles irá realizar a parceria naquele ciclo específico. As parcerias são estabelecidas simultaneamente, ciclo a ciclo da simulação, sendo que, sempre que necessário, a estrutura dos agentes é modificada¹, e os custos e benefícios de cada agente são computados. A simulação termina quando não houver mais parcerias possíveis de serem estabelecidas, pois os agentes só podem realizar uma ação uma única vez durante uma simulação.

¹Por exemplo, se um agente estabeleceu uma parceria para um objetivo num determinado ciclo, ele não tentará mais atingir este objetivo nos ciclos subseqüentes.

Numa simulação deste tipo, em princípio o resultado esperado é que os hedonistas tenham o maior ganho bruto, enquanto os utilitaristas tenham o maior ganho líquido, e os avaros tenham os menores custos. Em segunda instância, espera-se que os hedonistas se comportem de maneira menos flexível dos que os utilitaristas. Estudos anteriores mostraram que os hedonistas obtêm resultados comparáveis aos dos utilitaristas somente em grandes populações, justamente porque os hedonistas não são indiferentes ao contexto: quanto maior o número de agentes, maior a sua chance de encontrarem um parceiro ideal. Curiosamente, em estudos anteriores (CONTE, 1997), as duas estratégias se mostraram igualmente benéficas em ambientes com grandes populações. Através dos resultados obtidos no simulador Partnet, os estudos anteriores foram confirmados, pois pode-se observar que os hedonistas efetivamente se beneficiam quando há mais objetivos na sociedade. Em uma primeira análise, percebe-se que os hedonistas são mais rígidos do que os utilitaristas, mas utilizam uma estratégia social mais neutra, menos anti-social, uma vez que se preocupam exclusiavamente com o que recebem, sem se preocupar com o que gastam. Dessa forma, os hedonistas serão mais requisitados como parceiros, e sociedades heterogêneas, nos quais seja significativa a presença de hedonistas, apresentam um grau de benefício maior do que as compostas, por exemplo, exclusivamente de utilitaristas.

O simulador Partnet original foi escrito em 1997, utilizando a linguagem C. Sua utilização é extremamente difícil para usuários sem algum conhecimento prévio de programação.

O simulador \mathcal{P} ARTNET⁺ se propõe a efetuar algumas extensões ao simulador original, que podem ser divididas em duas classes:

- 1. extensões *estruturais*, relativas ao projeto e implementação do sistema e à sua interface com os usuários, tais como:
 - (a) adoção de uma filosofia de projeto do sistema orientada a objetos, utilizando-se a linguagem UML para descrever suas classes;
 - (b) nova implementação na linguagem JAVA, permitindo a execução do sistema também em navegadores, através de "applets";
 - (c) desenvolvimento de uma interface homem máquina mais amigável, com um sistema de janelas e elementos gráficos;

- (d) gestão da persistência e suporte à interoperabilidade, através da utilização da linguagem XML para armazenamento e recuperação dos resultados de simulação;
- 2. extensões funcionais, relativas às funções disponíveis no sistema, tais como:
 - (a) extensão da definição de parcerias para mais de 2 (dois) agentes, com a utilização da noção de plano, como descrita na seção 5.1.4;
 - (b) possibilidade de deteceção de parcerias caracterizadas por relações de dependência circulares, como descritas na seção 5.2.1;
 - (c) adoção de um nível de visibilidade para os agentes da sociedade, possibiltando simular a presença de agentes intermediários ("brokers").

As extensões estuturais já encontram-se finalizadas, sendo que a figura 10.1 apresenta a interface principal do simulador, bem como uma janela com os resultados, em forma gráfica, de uma simulação.

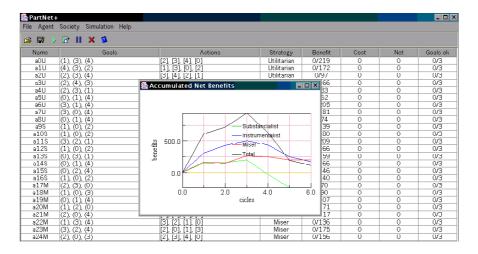


Figura 10.1: Interface do simulador \mathcal{P} ARTNET⁺.

A interface permite criar sociedades aleatórias com extrema facilidade, bastando passar os parâmetros adequados de criação, tais como: (i) número máximo de ações na sociedade; (ii) número máximo de objetivos na sociedade; (iii) número de ações por agente; (i) número de objetivos por agente; (iv) quantidade de agentes de cada um dos 3 (três) tipos possíveis (substancialistas, utilitários e avaros).

O simulador \mathcal{P} ARTNET⁺ permite ainda a geração de 3 (três) tipos de gráficos de saída: (i) o gráfico de benefício líquido acumulado por ciclos de simulação, que é mostrado na figura 10.1; (ii) o gráfico da evolução da média do benefício líquido entre a população e (iii) o gráfico do desvio padrão do benefício líquido, cuja forma é mostrada na figura 10.1.

10.2 Aplicação \mathcal{O} PTIMA

Máquinas insersoras de componentes são utilizadas na indústria eletrônica moderna para a montagem automática de placas de circuito impresso. Essas placas formam a parte principal dos equipamentos eletrônicos tanto de entretenimento (tais como rádios, televisores, vídeo cassetes), como de informática (tais como microcomputadores, monitores de vídeo, máquinas de FAX, etc.). A fase de montagem automatizada é uma área onde existem oportunidades para se obter potencialmente grandes economias. No Brasil, em particular, as máquinas de montagem automática (das quais as máquinas insersoras são um caso particular) têm preço relativamente alto, o que torna o seu custo de operação uma componente importante na matriz de custos de produção dos equipamentos eletrônicos. O objetivo da otimização desse processo é minimizar o tempo total de inserção, aumentando assim a utilização da máquina insersora.

O sistema Optima (RABAK, 1999; RABAK; SICHMAN, 2001) utiliza um time de agentes assíncronos (A-Teams) para resolver o problema de otimização do processo de inserção automática de componentes eletrônicos em placas de circuitos impressos. O sistema foi desenvolvido para otimizar o processo de uma máquina insersora particular², composta pelos seguintes componentes:

- 1. uma cabeça de inserção de componentes;
- 2. um mandril que se move entre o ponto de retirada de componentes, que é fixo na máquina, e o ponto de inserção na área da placa, transportando o componente do escaninho para a cabeça de inserção;
- 3. dois conjuntos de escaninhos para alojar os componentes;

²Modelo Panasert AVK, fabricada pela Panasonic.

4. uma mesa XY para movimentar a placa de circuito impresso sob a cabeça de inserção.

Esta máquina é caracterizada como sendo do tipo concorrente, fazendo com que seja possível movimentar a mesa XY e os escaninhos ao mesmo tempo. Uma unidade de controle comanda todo o conjunto, interpretando o programa de inserção para cada placa. Para o usuário, programar a insersora significa gerar um plano de atribuição de componentes a cada escaninho, e um programa de inserção, indicando em que posição da placa cada componente vai ser inserido. O programa deve incluir também a largura da dobra que o componente deve sofrer, a altura de inserção, o giro (em múltiplos de 90 graus, apenas) e se necessário um offset para ângulos não múltiplos de 90 graus.

Analisando-se a operação da máquina insersora, observa-se que a seqüência de inserção influi no tempo total de inserção, uma vez que uma escolha infeliz pode levar a mesa posicionadora a fazer grandes movimentos entre inserções seguidas, onerando o tempo total de inserção. Por outro lado, vê-se que a atribuição dos tipos de componentes aos escaninhos para uma dada seqüência de inserção também influi no tempo total, pois se entre duas inserções o escaninho anterior àquele contendo o componente a inserir estiver muito afastado deste último, a insersora terá seu tempo de inserção dominado pelo tempo de troca de escaninhos. Como o conjunto de escaninhos é o bloco (móvel) com maior massa na máquina insersora, é de se esperar que esse movimento da máquina seja mais lento. Portanto, o que se deve buscar, em última análise, é o melhor "casamento" entre os tempos obrigatórios devidos aos deslocamentos da mesa posicionadora com os tempos obrigatórios devidos às trocas de escaninhos, de modo a se ter as operações concorrentes com tempos os mais próximos possíveis. Nota-se também que a escolha da melhor atribuição dos componentes aos escaninhos é condicionada pela seqüência de inserção na placa, e por outro lado a melhor seqüência de inserção pode ser diferente do mínimo percurso (planar) na placa entre os pontos de inserção devido ao custo de troca de escaninhos. A melhor seqüência de inserção é condicionada, portanto, tanto pela atribuição dos componentes aos escaninhos, quanto pela seqüência de inserção.

A determinação do menor percurso, ou seja, da melhor seqüência de operações levando-se em conta os tempos dos ciclos de operação da máquina pode ser for-

mulada como uma instância do problema do caixeiro viajante (TSP) (LAWLER et al., 1985), enquanto a atribuição dos componentes aos escaninhos que minimize o tempo total de inserção pode ser formulada como uma instância do problema de alocação quadrática (QAP) (LAWLER, 1963). Tanto um como outro problema são conhecidos pela dificuldade de solução devido à explosão combinatória das soluções. Esses problemas são caracterizados ainda pela ausência de algoritmos que sejam exatos e rápidos. Os mais rigorosos são lentos ou consomem muitos recursos computacionais, enquanto os heurísticos não são muito robustos, sendo sensíveis à instância do problema: dão bons resultados para alguns casos e falham fragorosamente em outros casos.

A solução adotada no sistema \mathcal{O} PTIMA foi adotar a técnica dos *times assíncronos* (SOUZA; TALUKDAR, 1993, apud RABAK 1999), ilustrada na figura 10.2.

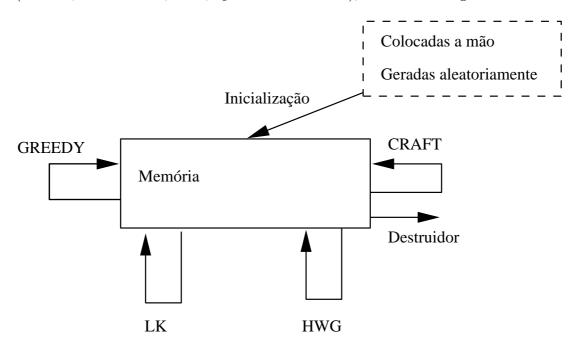


Figura 10.2: Exemplo de time assíncrono no sistema Optima (RABAK, 1999).

Um time assíncrono (A-Team) é um arranjo de agentes (representados por arestas na figura) que cooperam para atingir um objetivo desejado pelo seu projetista, sem que haja necessidade de uma coordenação prévia pré-estabelecida. A comunicação entre esses agentes se dá de forma indireta, através do ambiente, que é organizado em repositórios denominados memórias (representadas por retângulos na figura), onde são armazenadas soluções compartilhadas. O fato de não requererem coordenação permite que se organizem A-Teams onde os tempos

de processamento dos vários agentes sejam bem distintos, possibilitando assim que se possam combinar algoritmos mais velozes que produzem soluções mais fracas com aqueles que, embora tenham como saída soluções de melhor qualidade, são normalmente mais lentos.

Os agentes podem ser de dois tipos: construtores e destruidores. Construtores são aqueles que colocam novas soluções nas memórias. Destruidores apagam das memórias hipóteses de soluções, elminando assim aquelas menos promissoras. Os destruidores podem ainda ser empregados para eliminar padrões de produção de soluções indesejadas tais como seqüências de soluções repetidas.

Na implementação do sistema $\mathcal{O}PTIMA$, os agentes são programas que são lançados em função das escolhas efetuadas numa Interface Homem Máquina (IHM), conforme mostra a figura 10.3.

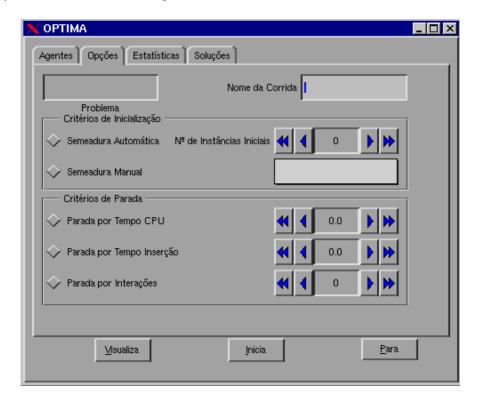


Figura 10.3: Interface com o usuário do sistema Optima (RABAK, 1999).

Neste painel, o usuário pode determinar o número de instâncias de solução que deseja que sejam criadas num processo de semeadura automática, as características do problema a tratar, etc. Embora a arquitetura do sistema $\mathcal{O}_{\text{PTIMA}}$ utilize uma única memória, para cada sessão de otimização cria-se um diretório

com nome diferente, de modo a possibilitar a análise de resultados de parametrizações diversas. O usuário pode iniciar uma sessão de otimização no sistema \mathcal{O} PTIMA usando dois critérios de inicialização: semeadura de soluções iniciais ou reutilização de soluções geradas numa sessão anterior. Quando é escolhida a semeadura de soluções na inicialização do A-Team, algumas soluções iniciais são geradas aleatoriamente, designando atribuições de tipos de componentes aos escaninhos e seqüências de inserção, de modo a criar uma população mínima para os agentes iniciarem seu trabalho. O sistema permite que se escolha o número inicial dessas soluções iniciais. Na reutilização, inicia-se o sistema com os resultados de uma sessão de otimização anterior, sem modificar os arquivos existentes na memória. Quanto ao critério de parada, podem ser utilizadas três estratégias distintas: (i) limite do tempo total de processamento do sistema; (ii) estabelecimento de um tempo total de inserção mínimo, que quando atingido para o processamento ou (iii) limite do número máximo de ciclos de resolução dos agentes.

O sistema está munido ainda de um visualizador, mostrado na figura 10.4, que permite verificar o percurso de inserção na placa de circuito impresso descrita pelo arquivo de entrada do problema para as várias soluções depositadas pelos agentes na memória do A-Team do sistema $\mathcal{O}_{\text{PTIMA}}$. Pode-se ainda verificar o tempo de inserção, o número de trocas de escaninhos, e as distâncias percorridas nas direções X e Y.

Uma primeira análise desses dados mostra que o tempo de inserção obtido com o sistema \mathcal{O} PTIMA foi inferior àquele fornecido pelo fabricante em todos os casos (RABAK, 1999; RABAK; SICHMAN, 2001). Além disso, observou-se que ao tratar algumas instâncias, o sistema \mathcal{O} PTIMA foi mais bem sucedido do que ao tratar outras, sendo que a redução média de tempo de inserção para todas as placas foi da ordem de 7,8%. Os resultados mostram ainda que a abordagem de combinar o QAP com o TSP consegue obter uma melhora adicional nos tempos de inserção de 4,7%, em média, em relação a uma abordagem que utiliza exclusivamente o TSP.

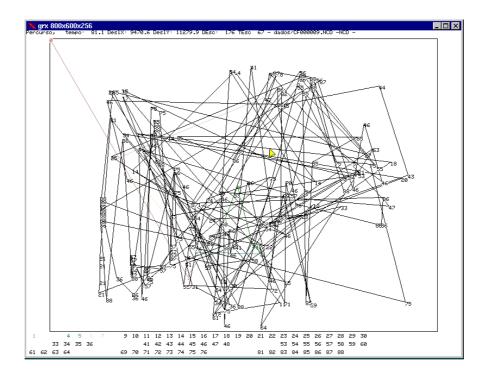


Figura 10.4: Visualizador do sistema Optima (RABAK, 1999).

10.3 Aplicações do projeto \mathcal{N} alamas

Segundo (NALAMAS, 1999), os trabalhos desenvolvidos no âmbito das linguagens de programação forneceram algum subsídio para o processamento de linguagens naturais (PLN). Estas últimas, porém, apresentam características que as tornam bem mais difíceis de processar, como a ambigüidade, a complexidade estrutural, as questões relacionadas ao significado e ao sentido.

O estudo da arquitetura dos sistemas existentes para PLN revela o predomínio das arquiteturas puramente seqüenciais, envolvendo os tratamentos associados aos diferentes níveis lingüísticos (léxico-morfológico, sintático, semântico, pragmático, etc), com todas as limitações de eficiência que lhe são inerentes.

O objetivo do projeto \mathcal{N} ALAMAS ³ ("Natural Language and Multi-Agent Systems") (PAIVA, 1997; LIMA et al., 1997; SILVA; ABRAHÃO; LIMA, 1998; CARVALHO et al., 1998) foi o de investigar uma alternativa a esta abordagem puramente seqüencial, através da utilização de SMA.

³Projeto financiado pelo CNPq, no âmbito do programa PROTEM-CC, processo CNPq 680081/95-0, coordenado pela PUC-RS, e envolvendo equipes da USP, UNICAMP, UFRGS, UFSC e Universidade Nova de Lisboa.

Em particular, foram desenvolvidos no âmbito deste projeto 2 (duas) aplicações, que experimentaram 2 (duas) diferentes abordagens de distribuição de processamento: uma abordagem lingüístico-cognitiva e uma abordagem léxico-estrutural, respectivamente através da utilização de SMA cognitivos e reativos, conforme descrito nas seções seguintes.

10.3.1 Abordagem cognitiva

Numa primeira abordagem, foi construído um SMA cognitivo (SILVA; ABRAHÃO; LIMA, 1998; CARVALHO et al., 1998) destinado a interpretar frases em português escrito. Os agentes lingüísticos da arquitetura proposta, mostrada na figura 10.5, têm um conhecimento do domínio lingüístico, e capacidades de percepção, ação e comunicação. Os agentes utilizam o modelo de descrição externa, discutido na seção 5.1.4, para representar as capacidades dos outros agentes.

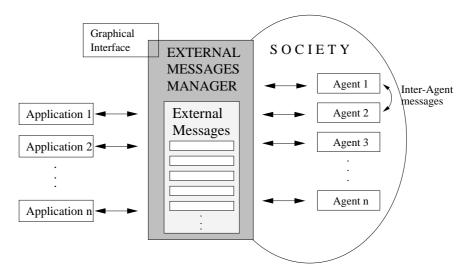


Figura 10.5: Abordagem cognitiva para PLN no projeto \mathcal{N} ALAMAS (SILVA; ABRAHÃO; LIMA, 1998).

Os agentes desenvolvidos foram associados aos níveis de processamento lingüístico: um agente léxico, um agente sintático e um agente semântico. Foram utilizados ainda protocolos de comunicação, como descritos na seção 3.6.4, para coordenar as suas atividades. Em termos organizacionais, tal sistema estaria associado a uma organização formal com foco estrutural, conforme a taxonomia apresentada na seção 4.4.

Tais agentes foram implementados na linguagem C, e integrados no ambi-

ente Masenv, descrito na seção 7.1. Segundo Silva, Abrahão e Lima (1998), a abordagem por SMA evita que hipóteses de frases menos promissoras sejam mantidas no espaço de hipóteses de solução, fazendo com que sejam descartadas mais rapidamente.

10.3.2 Abordagem reativa

Uma segunda abordagem, completamente distinta da primeira, é descrita em (PAIVA, 1997; CARVALHO et al., 1998). Nesta abordagem, os agentes são associados a palavras e sua interação se dá através de um mecanismo reativo, baseado em um campo de forças. Assim, palavras que normalmente seguem juntas numa frase são atraídas entre si, como por exemplo o artigo "O" e o adjetivo "velho" representados na figura 10.6. O reconhecimento de uma frase é feito de modo decentralizado, sendo que a frase é reconhecida quando todos os agentes estiverem ligados a pelo menos um outro agente. No exemplo mostrado na figura 10.6, existem duas ligações ("O velho" e "O palhaço") inicialmente geradas, sendo que neste caso a segunda é desfeita durante o processamento. Em termos organizacionais, tal sistema estaria associado a uma organização emergente, conforme discutido na seção 4.5.2. O sistema foi implementado na linguagem LISP.

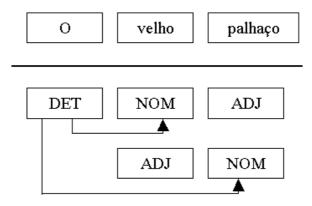


Figura 10.6: Abordagem reativa para PLN no projeto \mathcal{N} ALAMAS (PAIVA, 1997).

Uma ótima comparação entre as duas diferentes abordagens da utilização de SMA para PLN é apresentada em (CARVALHO et al., 1998). De forma resumida,

 $10.4 \quad Conclus\~oes$ 179

a abordagem reativa apresentou limitações inerentes à utilização de uma organização emergente, notadamente no que diz respeito ao custo computacional do procedimento de quebra de ligações entre as palavras. Por outro lado, apesar da abordagem cognitiva apresentar resultados promissores quanto à filtragem de hipóteses de frase incorretas, os resultados foram limitados pela quantidade de agentes desenvolvidos. Uma extensão dos trabalhos prevê o desenvolvimento de agentes que implementem a detecção de fenômenos linguísticos, tais como elipses e anáforas, bem como o projeto de interações e organizações que os levem em conta num procedimento de PLN.

10.4 Conclusões

A figura figura 10.7 mostra as aplicações \mathcal{P} ARTNET⁺, \mathcal{O} PTIMA e as 2 (duas) implementações desenvolvidas no projeto \mathcal{N} ALAMAS classificadas segundo os eixos básicos propostos na seção 2.3.

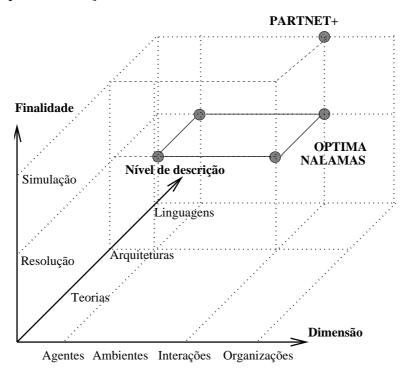


Figura 10.7: Classificação das outras aplicações multiagentes.

Em relação à sua finalidade, enquanto o sistema \mathcal{P} artnet⁺ tem uma finalidade de simulação social, os sistemas \mathcal{O} PTIMA e as aplicações do projeto \mathcal{N} Alamas adotam uma perspectiva de resolução social. O sistema \mathcal{P} artnet⁺

10.4 Conclusões 180

se situa no nível de descrição de linguagens e aborda exclusivamente a dimensão de interação entre agentes da metodologia Vogais. Já os sistemas \mathcal{O} PTIMA e as aplicações do projeto \mathcal{N} ALAMAS utilizam 2 (dois) níveis de descrição: arquitetura (para os agentes que os constituem) e linguagens. Quanto às dimensões da metodologia Vogais, ambos os sistemas abordam tanto a dimensão referente aos agentes quanto a dimensão referente às suas interações. Em particular, o sistema que adota a abordagem reativa no projeto \mathcal{N} ALAMAS utiliza um método de coordenação reativo, descrito na seção 3.5.4. O sistema \mathcal{O} PTIMA, apesar de ser um SMA com um nível baixo de granularidade, conforme descrito na seção 2.2.2, utiliza também um método de coordenação tipicamente reativo, através de sinais no ambiente.

Finalmente, no que diz respeito à representação das interações entre agentes, discutida na seção 3.3, enquanto na aplicação com abordagem cognitiva do projeto $\mathcal{N}_{\text{ALAMAS}}$ o modelo de interação adotado é intencional, na aplicação com abordagem reativa do mesmo projeto, bem como nos sistemas $\mathcal{O}_{\text{PTIMA}}$ e $\mathcal{P}_{\text{ARTNET}^+}$ considera-se um modelo não intencional, pois um observador externo é responsável por identificar as possíveis interações entre os agentes.

11 Conclusões

As teorias e aplicações sociais e organizacionais apresentadas neste trabalho permitem validar a idéia de que agentes cognitivos autônomos, imersos num SMA aberto, podem aumentar sua eficiência e adaptabilidade a alterações no ambiente, caso tenham a capacidade de representar explicitamente e explorar, através de mecanismos de raciocínio adequados, as capacidades de outros agentes e as eventuais organizações em que estejam envolvidos.

Foi demonstrado também que as dimensões entre as interações e as organizações de agentes formam um círculo virtuoso. Por um lado, as interações entre os agentes podem eventualmente criar organizações dinâmicas, denominadas coalizões, que propiciam uma ação coletiva. Por outro lado, caso tal ação coletiva deva se repetir com freqüência, torna-se mais adequado criar organizações formais, que limitem as suas interações entre os agentes, garantindo deste modo que estes atinjam seus objetivos globais de modo otimizado.

No que diz respeito às perspectivas futuras, vislumbram-se 3 (três) possíveis extensões para novas atividades de pesquisa.

Em primeiro lugar, deve-se notar que os mecanismos de raciocínio social e organizacional apresentados neste trabalho são utilizados de maneira isolada. Nas organizações humanas reais, este claramente não é o caso, pois as relações formais e informais co-existem num mesmo ambiente, sendo que a existência das últimas têm um grande fator de impacto no desempenho das organizações, como mostra o estudo realizado por Huberman e Hogg (1994). Assim, um modelo mais completo deveria levar em conta a inter-relação entre estes fenômenos. Um primeiro estudo preliminar neste sentido é apresentado em (SICHMAN; CONTE, 1998), onde se mostram os efeitos que o desempenho de um papel acarretam nas atitudes mentais pessoais de um agente e vice versa.

10 Conclusões 182

Como conseqüência desta perspectiva, uma outra extensão natural seria a concepção de um novo modelo de resolução cooperativa de problemas (WOOLDRID-GE; JENNINGS, 1994), discutido na seção 3.4, onde os resultados das interações entre os agentes pudessem modificar a própria estrutura organizacional. A proposta de reorganização utilizada no modelo \mathcal{M} OISE⁺, e discutida na seção 6.2.6, constitui-se claramente num primeiro passo nesta direção.

Finalmente, uma terceira perspectiva diz respeito à adoção de um novo mecanismo de raciocínio normativo em agentes autônomos. Noções como reputação ("reputation") e confiança ("trust") têm obtido cada vez mais importância no estudo de agentes autônomos, como bem mostram Conte e Paolucci (2002). Tais noções permitem, por exemplo, que um agente possa avaliar se vale a pena transgredir certas normas em algumas ocasiões, mesmo às custas de uma possível diminuição de sua reputação frente a outros agentes. Mecanismos como estes serão, com certeza, cada vez mais importantes em aplicações abertas e decentralizadas, como por exemplo em comércio eletrônico.

Anexo A – Descrição Formal do Modelo \mathcal{S}_{RM}

Nas seções abaixo, apresenta-se a descrição formal do modelo SRM descrito na seção 5.1, tal como originalmente apresentado em (SICHMAN, 1995; SICHMAN; DEMAZEAU, 2001).

A.1 LINGUAGEM INTERNA

Supõe-se que os agentes utilizem internamente uma linguagem de 1a. ordem. Por convenção, como se deseja deduzir propriedades sobre objetivos, planos, ações e recursos dos agentes, as variáveis utilizadas serão tipadas. Utilizar-se-á então a seguinte convenção para as variáveis: $\{i, j, k\}$ denotam agentes, $\{g, g'\}$ denotam objetivos, p denota planos, q denota ações e q denota recursos.

A.1.1 NOÇÕES BÁSICAS

A.1.1.1 NOÇÕES PRIMITIVAS

- achieves(p, g): o plano p se bem executado atinge o objetivo g
- $uses_a(p, a)$: o plano p utiliza a ação a
- $uses_r(p, r)$: o plano p utiliza o recurso r
- diff(g, g'): os objetivos $g \in g'$ são distintos
- $is_q(i,g)$: o agente i tem o objetivo g
- $is_a(i, a)$: o agente i sabe executar a ação a

- $is_r(i,r)$: o agente i tem controle sobre o recurso r
- $is_p(i, p)$: a agente i tem o plano p em seu repertório

A.1.1.2 NOÇÕES AUXILIARES

$$is_plan(i, g, p) \iff is_p(i, p) \land achieves(p, g)$$
 (A.1)

$$has_plans(i, g) \Leftrightarrow \exists p \ is_plan(i, g, p)$$
 (A.2)

$$needs_a(i, p, a) \Leftrightarrow uses_a(p, a) \land \neg is_a(i, a)$$
 (A.3)

$$has_all_a(i, p) \Leftrightarrow \forall a(uses_a(p, a) \Rightarrow is_a(i, a))$$
 (A.4)

$$needs_r(i, p, r) \Leftrightarrow uses_r(p, r) \land \neg is_r(i, r)$$
 (A.5)

$$has_all_r(i, p) \Leftrightarrow \forall r(uses_r(p, r) \Rightarrow is_r(i, r))$$
 (A.6)

A.1.1.3 NOÇÕES SOBRE PLANOS

$$aut_plan_a(i, g, p, k) \Leftrightarrow is_plan(k, g, p) \land has_all_a(i, p)$$
 (A.7)

$$aut_plan_r(i, g, p, k) \Leftrightarrow is_plan(k, g, p) \land has_all_r(i, p)$$
 (A.8)

$$dep_plan_a(i, j, g, p, k) \Leftrightarrow is_p(k, p) \land \exists \ a \ basic_dep_a(i, j, g, p, a)$$
 (A.9)

$$dep_plan_r(i, j, g, p, k) \Leftrightarrow is_p(k, p) \land \exists r \ basic_dep_r(i, j, g, p, r)$$
 (A.10)

onde:

$$basic_dep_a(i,j,g,p,a) \ \Leftrightarrow \ achieves(p,g) \land needs_a(i,p,a) \land is_a(j,a) \ (A.11)$$

$$basic_dep_r(i, j, g, p, r) \Leftrightarrow achieves(p, g) \land needs_r(i, p, r) \land is_r(j, r)$$
 (A.12)

A.1.2 NOÇÕES SOBRE AUTONOMIA

$$aut_a(i, g, k) \iff is_a(i, g) \land has_plans(k, g) \land \exists p \ aut_plan_a(i, g, p, k)$$
 (A.13)

$$aut_r(i, g, k) \iff is_q(i, g) \land has_plans(k, g) \land \exists p \ aut_plan_r(i, g, p, k)$$
 (A.14)

$$aut_s(i, g, k) \Leftrightarrow aut_a(i, g, k) \land aut_r(i, g, k)$$
 (A.15)

A.1.3 RELAÇÕES DE DEPENDÊNCIA

A.1.3.1 DEPENDÊNCIA BÁSICA

$$dep_a(i, g, k) \Leftrightarrow is_g(i, g) \land has_plans(k, g) \land \neg \exists p \ aut_plan_a(i, g, p, k) \ (A.16)$$

$$dep_r(i, g, k) \Leftrightarrow is_g(i, g) \land has_plans(k, g) \land \neg \exists p \ aut_plan_r(i, g, p, k) \ (A.17)$$

$$dep_s(i, g, k) \Leftrightarrow dep_a(i, g, k) \lor dep_r(i, g, k)$$
 (A.18)

$$dep_on_a(i, j, g, k) \Leftrightarrow dep_a(i, g, k) \land \exists p \ dep_plan_a(i, j, g, p, k)$$
 (A.19)

$$dep_on_r(i, j, g, k) \Leftrightarrow dep_r(i, g, k) \land \exists p \ dep_plan_r(i, j, g, p, k)$$
 (A.20)

$$dep_on_s(i, j, g, k) \Leftrightarrow dep_on_a(i, j, g, k) \lor dep_on_r(i, j, g, k)$$
(A.21)

A.1.3.2 Dependência mútua e recíproca

$$MD(i, j, g, k) \Leftrightarrow dep_on_a(i, j, g, k) \land dep_on_a(j, i, g, k)$$
 (A.22)

$$RD(i, j, g, g', k) \Leftrightarrow dep_on_a(i, j, g, k) \wedge dep_on_a(j, i, g', k) \wedge diff(g, g')$$
(A.23)

No restante do modelo formal, para efeito de simplificação, serão utilizadas apenas as noções de dependência e autonomia de ações, sendo omitido o subscrito, ou seja, utilizar-se-á $dep_on(i, j, g, k)$ como sinônimo de $dep_on_a(i, j, g, k)$.

A.1.4 SITUAÇÕES DE OBJETIVO

$$NG(i,g) \Leftrightarrow \neg is_g(i,g)$$
 (A.24)

$$NP(i,g) \Leftrightarrow is_g(i,g) \land \neg has_plans(i,g)$$
 (A.25)

$$AUT(i,g) \Leftrightarrow aut_a(i,g,i)$$
 (A.26)

$$DEP(i,g) \Leftrightarrow dep_a(i,g,i)$$
 (A.27)

A.1.5 SITUAÇÕES DE DEPENDÊNCIA

$$IND(i, j, g) \Leftrightarrow DEP(i, g) \land \neg dep_on_a(i, j, g, i)$$
 (A.28)

$$LBMD(i, j, g) \Leftrightarrow MD(i, j, g, i) \land \neg MD(i, j, g, j)$$
 (A.29)

$$MBMD(i, j, g) \Leftrightarrow MD(i, j, g, i) \land MD(i, j, g, j)$$
 (A.30)

$$LBRD(i, j, g, g') \Leftrightarrow RD(i, j, g, g', i) \land \neg RD(i, j, g, g', j)$$
 (A.31)

$$MBRD(i, j, g, g') \Leftrightarrow RD(i, j, g, g', i) \land RD(i, j, g, g', j)$$
 (A.32)

$$UD(i, j, g) \Leftrightarrow \frac{dep_on_a(i, j, g, i) \land}{\neg \exists g' (is_g(j, g') \land dep_on_a(j, i, g', i))}$$
(A.33)

A.1.6 Planos realizáveis

$$available_a(a) \iff \exists i \ is_a(i,a)$$
 (A.34)

$$available_r(r) \Leftrightarrow \exists i \ is_r(i,r)$$
 (A.35)

$$feasible_a(p) \Leftrightarrow \forall a \ (uses_a(p, a) \Rightarrow available_a(a))$$
 (A.36)

$$feasible_r(p) \Leftrightarrow \forall r (uses_r(p, r) \Rightarrow available_r(r))$$
 (A.37)

$$feasible_s(p) \Leftrightarrow feasible_a(p) \wedge feasible_r(p)$$
 (A.38)

A.1.7 Objetivos atingíveis

$$achievable_a(g, p) \Leftrightarrow achieves_a(p, g) \land feasible_a(p)$$
 (A.39)

$$achievable_r(g, p) \Leftrightarrow achieves_q(p, g) \land feasible_r(p)$$
 (A.40)

$$achievable_s(g, p) \Leftrightarrow achievable_a(g, p) \land achievable_r(g, p)$$
 (A.41)

$$achievable(i,g) \Leftrightarrow \begin{array}{l} AUT(i,g) \lor (DEP(i,g) \land \\ \exists \ p \ (is_p(i,p) \land achievable_a(g,p)) \end{array}$$
 (A.42)

A.2 LINGUAGEM EXTERNA

Para exprimir a linguagem externa dos agentes, utilizou-se o modelo dedutivo de crenças descrito em (KONOLIGE, 1986). Para isto, definiu-se um conjunto de

operadores de crença $\{\mathbf{B_i}, \mathbf{B_j}, \dots\}$ tal que a fórmula $\mathbf{B_i}$ ϕ , descrita na linguagem externa, significa que a fórmula ϕ , descrita na linguagem interna, pertence à base de crenças do agente i.

A.2.1 NOÇÕES BÁSICAS

$$Comp_q(i,j,k) \Leftrightarrow \forall g(\mathbf{B_i} \ is_q(k,g) \Leftrightarrow \mathbf{B_i} \ is_q(k,g))$$
 (A.43)

$$Comp_a(i, j, k) \Leftrightarrow \forall a(\mathbf{B_i} \ is_a(k, a) \Leftrightarrow \mathbf{B_j} \ is_a(k, a))$$
 (A.44)

$$Comp_r(i,j,k) \Leftrightarrow \forall r(\mathbf{B_i} \ is_r(k,r) \Leftrightarrow \mathbf{B_i} \ is_r(k,r))$$
 (A.45)

$$Comp_p(i,j,k) \Leftrightarrow \forall p(\mathbf{B_i} \ is_p(k,p) \Leftrightarrow \mathbf{B_j} \ is_p(k,p))$$
 (A.46)

A.2.2 Descrições externas compatíveis

$$Comp(i, j, k) \Leftrightarrow \frac{Comp_g(i, j, k) \wedge Comp_a(i, j, k) \wedge}{Comp_r(i, j, k) \wedge Comp_p(i, j, k)}$$
(A.47)

A.2.3 HIPÓTESE DE COMPATIBILIDADE DE DESCRIÇÃO EXTERNA

$$Ext_c(i,j) \Leftrightarrow Comp(i,j,i) \wedge Comp(i,j,j)$$
 (A.48)

A.2.4 Inconsistência no nível de agência

$$\mathbf{B_i} \ MBMD(i, j, g) \land \neg \mathbf{B_j} \ MBMD(j, i, g) \Rightarrow \neg Ext_c(i, j) \quad (A.49)$$

$$\mathbf{B_i} \ MBRD(i, j, g, g') \land \neg \mathbf{B_j} \ MBRD(j, i, g', g) \Rightarrow \neg Ext_c(i, j)$$
 (A.50)

$$\mathbf{B_i} \ LBMD(i, j, g) \wedge \mathbf{B_i} \ LBMD(j, i, g) \Rightarrow \neg Ext_c(i, j)$$
 (A.51)

$$\mathbf{B_i} \ LBRD(i, j, g, g') \wedge \mathbf{B_j} \ LBRD(j, i, g', g) \ \Rightarrow \ \neg Ext_c(i, j) \ (A.52)$$

Anexo B – Descrição Formal do Modelo \mathcal{S}_{RM}^+

Nas seções abaixo, apresenta-se a descrição formal do modelo \mathcal{S}_{RM^+} descrito na seção 5.2, tal como originalmente apresentado em (CONTE; SICHMAN, 2002; SICHMAN; CONTE, 2002).

B.1 Grafos de dependência

Um grafo de dependência $DPG = (V(DPG), E(DPG), \psi_{DPG})$ é um grafo orientado 4-partite com as seguintes propriedades:

- 1. o conjunto de vértices $V(DPG) = V_{ag}(DPG) \cup V_g(DPG) \cup V_p(DPG) \cup V_a(DPG)$ é formado pela união dos seguintes subconjuntos disjuntos de vértices:
 - (a) $V_{ag}(DPG) = \{ag_1, ag_2, \dots, ag_n\}$ é o conjunto de agentes;
 - (b) $V_g(DPG) = \{g_1, g_2, \dots, g_n\}$ é o conjunto de objetivos que os agentes desejam atingir;
 - (c) $V_p(DPG) = \{p_1, p_2, \dots, p_n\}$ é o conjunto de planos que os agentes podem utilizar para atingirem seus objetivos;
 - (d) $V_a(DPG) = \{a_1, a_2, \dots, a_n\}$ é o conjunto de ações que os agentes são capazes de realizar;
- 2. o conjunto E(DPG) é o conjunto de arestas;
- 3. a função de incidência $\psi_{DPG}: E(DPG) \mapsto V(DPG) \times V(DPG)$ é definida conforme abaixo:

- (a) $\psi_{DPG}(e) = (ag_i, g_j)$ associa uma aresta e ao par ordenado de vértices (ag_i, g_j) , com $ag_i \in V_{ag}(DPG)$ e $g_j \in V_g(DPG)$, e representa o fato de que o agente ag_i tem o objetivo g_i ;
- (b) $\psi_{DPG}(e) = (g_i, p_j)$ associa uma aresta e ao par ordenado de vértices (g_i, p_j) , com $g_i \in V_g(DPG)$ e $p_j \in V_p(DPG)$, e representa o fato de que o objetivo g_i pode ser alcançado através da execução do plano p_j ;
- (c) $\psi_{DPG}(e) = (p_i, a_j)$ associa uma aresta e ao par ordenado de vértices (p_i, a_j) , com $p_i \in V_p(DPG)$ e $a_j \in V_a(DPG)$, sendo que existe um agente $ag_k \in V_{ag}(DPG)$, um objetivo $g_l \in V_g(DPG)$ e arestas e_1 e $e_2 \in E(DPG)$ tal que $\psi_{DPG}(e_1) = (ag_k, g_l)$ e $\psi_{DPG}(e_2) = (g_l, p_i)$, e representa o fato de que o plano p_i utiliza a ação a_j , e o agente ag_k não é capaz de realizá-la;
- (d) $\psi_{DPG}(e) = (a_i, ag_j)$ associa uma aresta e ao par ordenado de vértices (a_i, ag_j) , com $a_i \in V_a(DPG)$ e $ag_j \in V_{ag}(DPG)$, e representa o fato de que a ação a_i pode ser realizada pelo agente ag_j .

B.2 Grafos de dependência reduzidos

Um grafo de dependência reduzido é um grafo de dependência onde cada agente tem apenas um único objetivo a atingir e um único plano para atingí-lo. Desta forma, a sua definição fica simplificada da seguinte forma:

- 1. o conjunto de vértices V(DPG) fica reduzido à união dos subsconjuntos que representam agentes e ações, ou seja, $V(DPG) = V_{ag}(DPG) \cup V_a(DPG)$;
- 2. a função de incidência $\psi_{DPG}: E(DPG) \mapsto V(DPG) \times V(DPG)$ fica restrita às associações de vértices nestes dois conjuntos, ou seja, composta por elementos dos tipos $\psi_{DPG}(e) = (ag_i, a_i)$ ou $\psi_{DPG}(e) = (a_i, ag_i)$;
- 3. define-se adicionalmente a função $\rho_{DPG}: E(DPG) \mapsto V_g(DPG) \cup NULL$, que identifica o objetivo g_k para o qual um agente ag_i depende da ação a_i :
 - (a) $\rho_{DPG}(e) = g_k$, se $\psi_{DPG}(e)$ é da forma (ag_i, a_j) , onde g_k é o objetivo para o qual ag_i necessita da ação a_j ;
 - (b) $\rho_{DPG}(e) = NULL$, se $\psi_{DPG}(e)$ é da forma (a_i, ag_j) .

B.3 AMONG-DEPENDÊNCIA

Dado um grafo reduzido de dependência DPG, ocorre uma AMONG-dependência entre um subconjunto de seus agentes se e somente se existir um subgrafo $G \subseteq DPG$ que contenha um ciclo de comprimento maior ou igual a 4 (quatro).

B.3.1 AMONG-DEPENDÊNCIA E OU-DEPENDÊNCIA

B.3.1.1 Desigualdade em AMONG-dependência

Quando um agente ag_i , que pertence a um grupo de agentes $Ag_j \subseteq V_{ag}(DPG)$ relacionados por uma AMONG-dependência tiver uma relação de OU-dependência, existirá pelo menos um vértice $a_i \in V_a(DPG)$ com $d_g^+(a_i) > d_g^-(a_i)$, onde $d_g^+(v)$ e $d_g^+(v)$ representam respectivamente os graus de incidência positivo e negativo do vértice $v \in V(DPG)$.

B.3.1.2 Incompatibilidade em AMONG-dependência

Uma AMONG-dependência é dita incompatível quando existirem um agente $ag_i \in V_{ag}(DPG)$ e uma ação $a_j \in V_a(DPG)$ tal que: (i) $d_g^+(a_j) > d_g^-(a_j)$; (ii) $d_g^+(ag_i) < d_g^-(ag_i)$ e (iii) $\exists e \in E(DPG)$ tal que $\psi_{DPG}(e) = (ag_i, a_j)$.

B.3.2 AMONG-DEPENDÊNCIA E E-DEPENDÊNCIA

B.3.2.1 AMONG-DEPENDÊNCIA FRÁGIL

Uma AMONG-dependência é dita frágil quando existirem dois agentes ag_i e $ag_j \in V_{ag}(DPG)$ tais que: (i) $d_g^+(a_i) > d_g^-(a_i)$ e (ii) $d_g^+(ag_j) < d_g^-(ag_j)$.

B.3.2.2 AMONG-DEPENDÊNCIA FORTE

Uma AMONG-dependência é dita forte quando para todos os agentes $ag_i \in V_{ag}(DPG)$ tem-se que $d_g^+(a_i) = d_g^-(a_i) = n_i$.

B.4 GROUP-DEPENDÊNCIA

Uma GROUP-dependência é dita decentralizada quando para todos os agentes $ag_i \in V_{ag}(DPG)$ tem-se que $d_g^+(a_i) = d_g^-(a_i) = n$.

B.5 COLLECTIVE-DEPENDÊNCIA

Uma COLLECTIVE-dependência é uma GROUP-dependence na qual para todas as arestas $e \in E(DPG)$ tem-se que $\rho_{DPG}(e) = g$ ou $\rho_{DPG}(e) = NULL$.

Anexo C – Descrição Formal do Modelo *M*oise

Nas seções abaixo, apresenta-se a descrição formal do modelo organizacional \mathcal{M} OISE descrito na seção 6.1, tal como originalmente apresentado em (HANNOUN et al., 1998; HANNOUN et al., 2000; HANNOUN, 2002).

C.1 Definições preliminares

Sejam Any_g , Any_p , Any_p , Any_r , Any_{pr} e Any_{ac} respectivamente o conjunto de todos os objetivos, planos, ações, recursos, protocolos de comunicação e atos de fala. Cada elemento destes conjuntos é denotado respectivamente pelas variáveis g, p, a, r, pr e ac. Quando for necessário representar classes de elementos pertencentes a estes conjuntos, é utilizado um subscrito, por exemplo g_{prof} .

O conceito de ação instanciada, que designa uma ação e o seu conjunto associado de recursos, é representado pela n-upla $(a_i, r_{j_1}, r_{j_2}, ..., r_{j_n}) \in Any_a \times Any_r^n$.

C.1.1 Plano

Um plano p_i é definido formalmente como:

$$p_i \stackrel{\triangle}{=} \langle g_i, \langle K_{p_i}, V_{p_i}, T_{p_i}, d_{p_i}, F_{p_i} \rangle \rangle \tag{C.1}$$

onde g_i é o objetivo que o plano visa atingir, $\langle K_{p_i}, V_{p_i}, T_{p_i}, d_{p_i}, F_{p_i} \rangle$ é um AF tal que K_{p_i} é o conjunto de seus estados, V_{p_i} é o conjunto de suas arestas, que

representam as ações instanciadas ou sub-objetivos, $T_{p_i}: K_{p_i} \times V_{p_i} \Rightarrow K_{p_i}$ é uma função de transição, $d_{p_i} \in K_{p_i}$ é o estado inicial e $F_{p_i} \subset K_{p_i}$ é o conjunto de estados finais.

Para exprimir um plano como uma expressão regular, são utilizados os operadores «;», «|» e «*», que exprimem respectivamente a seqüência, escolha, e a repetição de ações.

Um exemplo típico de plano seria o seguinte:

$$p_1(g_1) = a_1(r_1, r_2); (a_2(r_3)^* | [g_{10}]); a_3.$$

Tal plano $p_1(g_1)$, destinado a atingir o objetivo g_1 , determina a execução em primeiro lugar da ação a_1 , que consome os recursos r_1 e r_2 . Posteriormente, solicita a execução da ação a_2 um número qualquer de vezes (inclusive 0), sendo que tal ação consome o recurso r_3 , ou então, alternativamente, tenta atingir o subobjetivo g_{10} . Finalmente, determina que se execute a ação a_3 , que não necessita de recurso algum para ser efetuada.

C.1.2 Protocolo de comunicação

Um protocolo de comunicação pr_i é definido formalmente como:

$$pr_i \stackrel{\triangle}{=} \langle K_{pr_i}, V_{pr_i}, T_{pr_i}, d_{pr_i}, F_{pr_i} \rangle$$
 (C.2)

sendo o protocolo especificado pelo AF $\langle K_{pr_i}, V_{pr_i}, T_{pr_i}, d_{pr_i}, F_{pr_i} \rangle$, tal que K_{pr_i} é o conjunto de estados do protocolo , V_{pr_i} é o conjunto de suas arestas, que representam os atos de fala, $T_{p_i}: K_{p_i} \times V_{p_i} \Rightarrow K_{p_i}$ é uma função de transição, $d_{p_i} \in K_{p_i}$ é o estado inicial e $F_{p_i} \subset K_{p_i}$ é o conjunto de estados finais.

Um protocolo também pode ser representado através de uma expressão regular. Um exemplo típico de protocolo de comunicação do gênero licitação seria o seguinte:

$$pr_1 = Request; (Refuse \mid (Accept; (Cancel \mid Confirm))).$$

C.2 Nível individual 194

C.2 NÍVEL INDIVIDUAL

C.2.1 Papel

$$ro = \{(t_i, m_i)/t_i \in T \in m_i \in M\}$$
 (C.3)

onde ro é um papel, M é o conjunto de missões e T é o conjunto de possíveis naturezas de uma missão $T = \{\mathbf{O}, \mathbf{P}\}$, associadas respectivamente às noções de obrigação (\mathbf{O}) e permissão (\mathbf{P}) .

C.2.2 Missão

Uma missão m_i é definida formalmente como:

$$m_i = \langle G_i, P_i, A_i, R_i \rangle \tag{C.4}$$

onde G_i , P_i , A_i e R_i são respectivamente os conjuntos de objetivos, planos, ações e recursos permitidos no contexto da missão. Cada um destes conjuntos pode conter a classe $Any_{g/p/a/r}$, quando qualquer elemento do conjunto pode ser utilizado, ou ao contrário ser vazio (denotado por \varnothing), quando nenhum elemento pode ser utilizado.

C.3 NÍVEL SOCIAL

C.3.1 RELAÇÃO ORGANIZACIONAL

Uma relação organizacional l_i é definida formalmente como:

$$l_i = \langle ro_s, ro_d, t_i, \mathcal{M}(ro_s), \mathcal{M}(ro_d), C_i \rangle$$
 (C.5)

sendo que a relação organizacional ocorre entre um papel fonte ro_s e um papel destino ro_d , de tipo $t_i \in COM, AUT, CON$ (comunicação, autoridade, conhecimento), e esta relação é válida somente no contexto do subconjunto de missões

C.4 Nível coletivo

 $\mathcal{M}(ro_s)$ do papel fonte e do subconjunto de missões $\mathcal{M}(ro_d)$ do papel destino. Cada um destes subconjuntos pode assumir o valor Any_m , que corresponde ao conjunto de todas as missões do respectivo papel. Finalmente, C_i respresenta um conjunto de restrições associadas à relação e depende do tipo da relação. Por exemplo, no caso de uma relação de comunicação, representa o conjunto de protocolos autorizados para a troca de mensagens entre os agentes.

C.4 NÍVEL COLETIVO

C.4.1 Grupo

Um grupo gr_i é definido formalmente como:

$$gr_i = \langle Ro_i, M_i, L_i \rangle$$
 (C.6)

onde Ro_i é um conjunto de papéis, M_i é um subconjunto de missões associadas a estes papéis e L_i é um subconjunto de relações organizacionais que conectam estes papéis no contexto deste subconjunto de missões.

C.5 ESTRUTURA ORGANIZACIONAL

Uma estrutura organizacional os_i é definida formalmente como:

$$os_i = \langle Ro_{os_i}, L_{os_i}, GR_{os_i} \rangle$$
 (C.7)

onde Ro_{os_i} é o conjunto de seus papéis, L_{os_i} é o conjunto de suas relações organizacionais e GR_{os_i} é o conjunto de seus grupos.

C.6 ENTIDADE ORGANIZACIONAL

Uma entidade organizacional oe_i é definida formalmente como:

$$oe_i = \langle os_k, IR_{oe_i}, IGR_{oe_i} \rangle.$$
 (C.8)

onde oe_i é instanciada da estrutura organizacional os_k , através dos pares agentepapel IR_{oe_i} , e contendo as instâncias de grupo IGR_{oe_i} .

C.6.1 Papéis instanciados

Um conjunto de papéis instanciados IR_{oe_i} é definido formalmente como:

$$IR_{oe_i} = \{(ag_j, ro_k)/is_agent(ag_j, oe_i) \text{ et } is_role(ro_k, os_k)\}.$$
 (C.9)

onde (ag_j, ro_k) é um par composto pelo agente ag_j e pelo papel ro_k , tal que o agente ag_j desempenha o papel ro_k .

C.6.2 Instância de grupo

Uma instância de grupo igr_l é definida formalmente como:

$$igr_l = \langle gr_i, eo_i, \{(ag_k, ro_l)\}\rangle \tag{C.10}$$

onde gr_i representa o grupo abstrato do qual igr_l é uma instância, eo_i é a entidade organizacional à qual igr_l pertence e $\{(ag_k, ro_l)\}$ é o subconjunto das duplas agente-papel da entidade organizacional que pertencem ao grupo.

Anexo D – Descrição Formal do Modelo \mathcal{M} oise⁺

Nas seções abaixo, apresenta-se a descrição formal do modelo organizacional \mathcal{M} OISE⁺ descrito na seção 6.2, tal como originalmente apresentado em (HÜBNER; SICHMAN; BOISSIER, 2002a; HÜBNER; SICHMAN; BOISSIER, 2002b; HÜBNER; SICHMAN; BOISSIER, 2002c).

D.1 ESPECIFICAÇÃO ORGANIZACIONAL

Definição D.1 (*Especificação Organizacional*) O conjunto de todas as especificações organizacionais (\mathcal{OS}) possíveis é formado por tuplas de aridade três:

$$\mathcal{OS} =_{def} \mathcal{SS} \times \mathcal{FS} \times \mathcal{DS}$$
 (D.1)

onde SS é o conjunto das especificações estruturais; FS é o conjunto das especificações funcionais; e DS é o conjunto das especificações deônticas.

D.2 ESPECIFICAÇÃO ESTRUTURAL

D.2.1 NÍVEL INDIVIDUAL: PAPÉIS

Definição D.2 (**Papel**) Um papel ρ é um identificador que pertence ao conjunto de identificadores de papéis \mathcal{R}_{ss} da especificação estrutural ($\rho \in \mathcal{R}_{ss}$).

Definição D.3 (*Papel abstrato*) O conjunto de todos os papéis abstratos é denotado por \mathcal{R}_{abs} sendo que a seguinte propriedade deve ser satisfeita

$$\mathcal{R}_{abs} \subset \mathcal{R}_{ss}$$
 (D.2)

Definição D.4 (**Herança**) O predicado $\rho \sqsubset \rho'$ é verdade se

- o papel ρ' é uma especialização do papel ρ e
- $\rho \in \mathcal{R}_{ss} \wedge \rho' \in \mathcal{R}_{ss}$.

As fórmulas seguintes são verdadeiras para o predicado de herança (anti-simetria e transitividade):

$$\rho \sqsubset \rho' \land \rho' \sqsubset \rho \quad \Rightarrow \quad \rho = \rho' \tag{D.3}$$

$$\rho \sqsubset \rho' \land \rho' \sqsubset \rho'' \quad \Rightarrow \quad \rho \sqsubset \rho'' \tag{D.4}$$

Definição D.5 (**Papel social**) Dado o papel abstrato ρ_{soc} , acrescentam-se as seguintes propriedades ao conjunto de papéis e à relação de herança:

$$\rho_{soc} \in \mathcal{R}_{ss}$$
 (D.5)

$$\forall \rho \in (\mathcal{R}_{ss} - \{\rho_{soc}\}) \ \rho_{soc} \sqsubset \rho \tag{D.6}$$

$$\nexists \rho \in \mathcal{R}_{ss} \ \rho \sqsubset \rho_{soc} \tag{D.7}$$

D.2.2 NÍVEL SOCIAL: LIGAÇÕES

Definição D.6 (Ligação) O predicado $link(\rho_s, \rho_d, t)$ é verdade se

- o papel ρ_s tem uma ligação do tipo t com o papel ρ_d e
- $\rho_s \in \mathcal{R}_{ss} \wedge \rho_d \in \mathcal{R}_{ss} \wedge t \in \mathcal{LT}$,

sendo que $\mathcal{LT} = \{acq, com, aut\}$ e as fórmulas seguintes são verdadeiras para o predicado de ligação:

$$(link(\rho_s, \rho_d, t) \land \rho_s \sqsubset \rho_s') \Rightarrow link(\rho_s', \rho_d, t)$$
 (D.8)

$$(link(\rho_s, \rho_d, t) \land \rho_d \sqsubset \rho_d') \Rightarrow link(\rho_s, \rho_d', t)$$
 (D.9)

$$link(\rho_s, \rho_d, aut) \Rightarrow link(\rho_s, \rho_d, com)$$
 (D.10)

$$link(\rho_s, \rho_d, com) \Rightarrow link(\rho_s, \rho_d, acg)$$
 (D.11)

As duas primeiras fórmulas determinam como as ligações se comportam quanto à herança e as duas últimas determinam que uma ligação de autoridade implica na

existência de uma ligação de comunicação que, por sua vez, implica na existência de uma ligação de conhecimento.

Definição D.7 (Compatibilidade) O predicado de $\rho_a \bowtie \rho_b$ é verdade se

- o papel ρ_a é compatível com o papel ρ_b e
- $\rho_a \in \mathcal{R}_{ss} \wedge \rho_b \in \mathcal{R}_{ss}$.

As fórmulas seguintes são verdadeiras para a relação de compatibilidade (reflexividade, transitividade e herdável):

$$\rho \bowtie \rho$$
 (D.12)

$$\rho \bowtie \rho' \land \rho' \bowtie \rho'' \quad \Rightarrow \quad \rho \bowtie \rho'' \tag{D.13}$$

$$\rho_a \bowtie \rho_b \land \rho_a \neq \rho_b \land \rho_a \sqsubset \rho' \implies \rho' \bowtie \rho_b \tag{D.14}$$

$$\rho \sqsubset \rho' \Rightarrow \rho' \bowtie \rho \tag{D.15}$$

D.2.3 NÍVEL COLETIVO: GRUPOS

Definição D.8 (*Especificação de Grupo*) Sendo \mathcal{GT} o conjunto de todas as especificações de grupo possíveis, uma especificação de grupo $gt \in \mathcal{GT}$ é representada pela tupla

$$gt = (\mathcal{R}, \mathcal{SG}, \mathcal{L}^{intra}, \mathcal{L}^{inter}, \mathcal{C}^{intra}, \mathcal{C}^{inter}, np, ng)$$
 (D.16)

sendo:

• \mathcal{R} o conjunto de papéis que podem ser assumidos em grupos gt (grupos criados a partir da especificação gt)

$$\mathcal{R} \subseteq \mathcal{R}_{ss} \tag{D.17}$$

• \mathcal{SG} o conjunto de sub-grupos possíveis para gt

$$\mathcal{SG} \subseteq \mathcal{GT} \tag{D.18}$$

- \mathcal{L}^{intra} e \mathcal{L}^{inter} os conjuntos das ligações em grupos gt;
- \mathcal{C}^{intra} e \mathcal{C}^{inter} os conjuntos das compatibilidades em grupos gt;

 $\bullet \ np: \mathcal{NP}$ um mapeamento parcial para a cardinalidade de papéis

$$\mathcal{NP} =_{\text{def}} \mathcal{R}_{ss} \to \mathbb{N} \times \mathbb{N} \tag{D.19}$$

• $ng: \mathcal{NG}$ um mapeamento parcial para a cardinalidade de sub-grupos

$$\mathcal{NG} =_{\mathsf{def}} \mathcal{SG} \to \mathbb{N} \times \mathbb{N} \tag{D.20}$$

Definição D 9 (Especificação de grupo raíz) O predicado isRootGr(gt) é verdade se a especificação de grupo gt não é sub-grupo de nenhuma outra especificação de grupo, isto é, é um grupo raíz:

$$isRootGr(gt) \Leftarrow \nexists g \in \mathcal{GT} \text{ tal que } gt \in g.\mathcal{SG}$$
 (D.21)

Definição D 10 (Especificação Estrutural) Sendo SS o conjunto de todas as especificações estruturais, uma especificação estrutural ss é representada por uma tupla

$$ss = (\mathcal{RG}, \mathcal{R}_{ss}, \square)$$
 (D.22)

onde:

• \mathcal{RG} é o conjunto de especificações de grupos raízes

$$\mathcal{RG} \subseteq \mathcal{GT} \tag{D.23}$$

$$\forall gt \in \mathcal{RG} \cdot isRootGr(gt) \tag{D.24}$$

- \mathcal{R}_{ss} é o conjunto de todos os papéis e
- \sqsubseteq é a relação de herança sobre os papéis de \mathcal{R}_{ss} .

O conjunto de todas as especificações de grupo, denotado por \mathcal{AG} ($\mathcal{AG} \subseteq \mathcal{GT}$), é dado pelos grupos definidos em \mathcal{RG} e, recursivamente, seus sub-grupos:

$$\mathcal{AG} = \bigcup_{g \in \mathcal{RG}} allSubGroups(g)$$
 (D.25)

$$\mathcal{AG} = \bigcup_{g \in \mathcal{RG}} allSubGroups(g)$$
(D.25)
$$allSubGroups(gt) = \{gt\} \cup \bigcup_{g \in gt.\mathcal{SG}} allSubGroups(g)$$
(D.26)

O conjunto de papéis abstratos é calculado a partir do conjunto de todos os

papéis menos os papéis que podem ser assumidos em algum grupo:

$$\mathcal{R}_{abs} = \mathcal{R}_{ss} - \bigcup_{\forall gt \in \mathcal{AG}} gt.\mathcal{R}$$
 (D.27)

D.3 ESPECIFICAÇÃO FUNCIONAL

D.3.1 METAS GLOBAIS

A cada meta global g é associado uma combinação dos seguintes valores:

- unsatisfied: a meta ainda não foi satisfeita, o mundo não está como representado em g;
- satisfied: a meta foi satisfeita, o mundo está como representado em g;
- *impossible*: é impossível satisfazer a meta, isto é, o estado do mundo representado por g nunca poderá ser alcançado;
- uncommitted: a meta n\(\tilde{a}\) possui qualquer agente comprometido em satisfaz\(\tilde{e}\)-la;
- committed: a meta possui pelo menos um agente comprometido em satisfazê-la;
- forbiden: a especificação funcional não permite que o agente comprometido com a meta tente satisfazê-la, isto é, no contexto atual, não há condições favoráveis para que a meta possa ser satisfeita. Por exemplo, a meta "entregar a documentação" é forbiden até que a documentação esteja toda preparada.
- permitted: a especificação funcional permite que o agente comprometido com a meta tente satisfazê-la.

O valor inicial de uma meta é (unsatified, uncommitted, forbiden) e, no decorrer do funcionamento, vai alterando este valor. Os três valores primeiros da lista acima são mutuamente exclusivos, bem como uncommitted e committed; e forbiden e permitted. Portando o valor de uma meta g deve pertencer ao conjunto \mathcal{GV} dado por

$$\mathcal{GV} = \{\{unsatisfied, satisfied, impossible\} \times \\ \{uncommitted, committed\} \times \\ \{forbiden, permitted\} \}$$

D.3.2 NÍVEL INDIVIDUAL: MISSÕES

Definição D.11 (Missão) Sendo \mathcal{M} o conjunto de todas as missões, uma missão $m \in \mathcal{M}$ é um conjunto de metas globais

$$m \subset \mathbb{P}(\mathcal{GV})$$
 (D.28)

D.3.3 NÍVEL COLETIVO: ESQUEMAS

Definição D.12 (**Esquema Social**) O conjunto de todos os esquemas é denotado por \mathcal{SCH} e um esquema sch é representado pela tupla

$$sch = (\mathcal{G}, \mathcal{P}, \mathcal{M}, mo, nm)$$
 (D.29)

onde:

- \mathcal{G} é o conjunto de metas do ES sch;
- P é o conjunto de planos que constrói a árvore de decomposição de metas do ES;
- \mathcal{M} é o conjunto de missões;
- $mo: \mathcal{M} \to \mathbb{P}(\mathcal{G})$ é uma função que determina o conjunto de metas de cada missão e
- $nm : \mathcal{M} \to \mathbb{N} \times \mathbb{N}$ determina o número (mínimo, máximo) de agentes que devem se comprometer em cada missão.

D.3.4 Preferência entre missões

Definição D.13 (**Preferência entre missões**) As preferências entre as missões são representadas por uma relação de ordem parcial \prec : ($\mathcal{M}_{fs} \times \mathcal{M}_{fs}$), sendo \mathcal{M}_{fs} o conjunto das missões definidas. Esta relação é

- anti-simétrica $(m_1 \prec m_2 \land m_2 \prec m_1 \Rightarrow m_2 = m_1)$ e
- transitiva $(m_1 \prec m_2 \land m_2 \prec m_3 \Rightarrow m_1 \prec m_3)$.

Definição D.14 (Especificação Funcional) A Especificação Funcional de uma organização é um conjunto (S) de ES e a relação de preferência entre as missões destes esquemas (PR). Portanto, uma EF fs do conjunto de todas as EF, denotado por FS, é dado pelo par

$$fs = (S, \mathcal{PR})$$
 (D.30)

onde:

$$S \subseteq \mathcal{SCH} \tag{D.31}$$

$$\mathcal{PR} \subseteq \mathcal{M}_{fs} \times \mathcal{M}_{fs} \tag{D.32}$$

O conjunto de todas as missões é dado por:

$$\mathcal{M}_{fs} = \bigcup_{sch \in \mathcal{S}} sch.\mathcal{M} \tag{D.33}$$

O conjunto de todas as metas é dado por:

$$\mathcal{G}_{fs} = \bigcup_{sch \in \mathcal{S}} sch.\mathcal{G} \tag{D.34}$$

D.4 ESPECIFICAÇÃO DEÔNTICA

Definição D.15 (*Especificação Deôntica*) A Especificação Deôntica de uma organização é representada pela tupla

$$(\mathcal{P}, \mathcal{O}) \tag{D.35}$$

onde:

 \bullet \mathcal{P} é um conjunto de permissões

$$\mathcal{P} \subseteq \mathcal{R}_{ss} \times \mathcal{M}_{fs} \times \mathbb{P}(Any) \tag{D.36}$$

• \mathcal{O} é um conjunto de obrigações

$$\mathcal{O} \subseteq \mathcal{R}_{ss} \times \mathcal{M}_{fs} \times \mathbb{P}(Any) \tag{D.37}$$

D.5 ENTIDADE ORGANIZACIONAL

Definição D.16 (*Entidade Organizacional*) Uma entidade organizacional é formada por uma finalidade que deve ser mantida por um conjunto de agentes que instanciam uma entidade. O estado de uma entidade em um instante t é denotado pela tupla

$$oe = (\gamma, os, \mathcal{A}, \mathcal{GI}, grType, subGr, agRole, \mathcal{SI}, scType, agMis, gState)^t$$
 (D.38) onde

- γ é a finalidade da entidade;
- os é a EO desta EnO;
- \mathcal{A} é o conjunto de identificadores dos agentes que pertencem a esta entidade;
- \mathcal{GI} é o conjunto dos grupos criados;
- $grType: \mathcal{GI} \to \mathcal{AG}_{ss}$ determina as especificações dos grupos em \mathcal{GI} ;
- $subGr: \mathcal{GI} \to \mathbb{P}(\mathcal{GI})$ determina os sub-grupos dos grupos;
- agRole: A → P(R_{ss} × GI) representa o conjunto de papéis que cada agente está assumindo no momento t (nota-se que um papel somente pode ser assumido no contexto de um grupo);
- \mathcal{SI} é o conjunto de ES criados;
- $scType: \mathcal{SI} \to \mathcal{S}_{fs} \times \mathbb{P}(\mathcal{GI})$ determina a especificação dos ES criados e também quais os grupos que criaram o ES;
- $agMis: \mathcal{A} \to \mathbb{P}(\mathcal{M}_{fs} \times \mathcal{SI})$ representa o conjunto de missões com que cada agente está comprometido e
- $gState: \mathcal{SI} \times \mathcal{G}_{fs} \rightarrow (GV)$ representa o estado das metas dos esquemas.

Para colocar a lista de referências no índice

Referências Bibliográficas

AGHA, G.; HEWITT, C. E. Concurrent programming using actors: Exploiting large-scale parallelism. In: BOND, A. H.; GASSER, L. (Ed.). *Readings in Distributed Artificial Intelligence*. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1986. p. 398–407.

ALBUQUERQUE, R. L. et al. KSACI: A handheld device infrastructure for agents communication. In: MEYER, J.-J. C.; TAMBE, M. (Ed.). *Intelligent Agents VIII (Proceedings of the 8th. International Workshop on Agent Theories, Architectures and Languages)*. Berlin, DE: Springer-Verlag, 2002, (Lecture Notes in Artificial Intelligence, v. 2333). p. 423–435.

AUSTIN, J. L. How to Do Things with Words. Oxford, UK: Oxford University Press, 1962.

AXELROD, R. The Evolution of Cooperation. New York: Basic Books, 1984.

BOISSIER, O. Problème du Contrôle dans un Système Integré de Vision. Utilisation d'un Système Multi-Agents. Tese (Thèse de Doctorat) — Institut National Polytechnique de Grenoble, Grenoble, France, January 1993.

BOISSIER, O. et al. *The Multi-Agent System Toolkit*. Saint-Etienne, France, 1998. Technical Report.

BOISSIER, O.; DEMAZEAU, Y. ASIC: An architecture for social and individual control and its application to computer vision. In: DEMAZEAU, Y.; MÜLLER, J.-P.; PERRAM, J. (Ed.). Pre-proceedings of the 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Odense, Denmark: [s.n.], 1994. p. 107–118.

BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. Plans and resource-bounded practical reasoning. *Computational Intelligence*, v. 4, p. 349–355, 1988.

BROOKS, R. A. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, v. 2, n. 1, p. 435–453, March 1986.

CARDOZO, E. *DPSK: A Kernel for Distributed Problem Solving*. Tese (PhD Thesis) — CAED, Carnegie Mellon University, Pittsburgh, PE, January 1987.

CARDOZO, E.; SICHMAN, J. S. *DPSK+P User's Manual - C++ Interface*. Campinas, Brasil, October 1992. Technical Report.

- CARDOZO, E.; SICHMAN, J. S.; DEMAZEAU, Y. Using the active object model to implement multi-agent systems. In: *Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence*. Boston, USA: IEEE Computer Society Press, 1993. p. 70–77.
- CARLEY, K.; GASSER, L. Computational organization theory. In: WEISS, G. (Ed.). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999. p. 299–330.
- CARVALHO, A. M. B. R. et al. Multi-agent systems for natural language processing. In: *Proceedings of the 2nd Ibero-American Workshop on DAI/MAS*. Toledo, Spain: [s.n.], 1998. p. 61–69.
- CASTELFRANCHI, C. Social power: A point missed in multi-agent, DAI and HCI. In: DEMAZEAU, Y.; MÜLLER, J.-P. (Ed.). *Decentralized A. I.* Amsterdam, NL: Elsevier Science Publishers B. V., 1990. p. 49–62.
- CASTELFRANCHI, C.; MICELLI, M.; CESTA, A. Dependence relations among autonomous agents. In: WERNER, E.; DEMAZEAU, Y. (Ed.). *Decentralized A. I. 3.* Amsterdam, NL: Elsevier Science Publishers B. V., 1992. p. 215–227.
- COHEN, P. R.; LEVESQUE, H. J. Intention = choice + commitment. In: *Proceedings of the 6th National Conference on Artificial Intelligence*. Seattle, WA: Morgan Kaufmann Publishers, Inc., 1987. p. 410–415.
- COHEN, P. R.; PERRAULT, C. R. Elements of a plan based theory of speech acts. *Cognitive Science*, v. 3, p. 177–212, 1979.
- CONTE, R. Diversity in Rationality: A Multi- Agent Perspective. 1997. Dagstuhl Seminar on Social Science Microsimulation.
- CONTE, R.; CASTELFRANCHI, C. Mind is not enough: Precognitive bases of social interaction. In: GILBERT, N. (Ed.). *Proceedings of 1992 Symposium on Simulating Societies*. Guildford, UK: [s.n.], 1992. p. 93–110.
- CONTE, R.; PAOLUCCI, M. Reputation in Artificial Societies: Social Beliefs for Social Order. Amsterdam, NL: Kluwer Academic Publishers, 2002.
- CONTE, R.; PEDONE, R. Finding the best partner: The PARTNET system. In: SICHMAN, J. S.; CONTE, R.; GILBERT, N. (Ed.). *Multi-Agent Systems and Agent-Based Simulation*. Berlin, DE: Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v. 1534), p. 156–168.
- CONTE, R.; SICHMAN, J. S. DEPNET: How to benefit from social dependence. Journal of Mathematical Sociology, v. 20, n. 2–3, p. 161–177, 1995.
- CONTE, R.; SICHMAN, J. S. Dependence graphs: Dependence within and between groups. *Computational & Mathematical Organization Theory*, v. 8, n. 2, p. 87–112, 2002.

- CONTE, R.; SICHMAN, J. S.; GILBERT, N. MAS and social simulation: A suitable commitment. In: SICHMAN, J. S.; CONTE, R.; GILBERT, N. (Ed.). *Multi-Agent Systems and Agent-Based Simulation*. Berlin, DE: Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v. 1534), p. 1–9.
- CONTE, R.; VENEZIANO, V.; CASTELFRANCHI, C. The computer simulation of partnership formation. *Computational & Mathematical Organization Theory*, v. 4, n. 4, p. 293–315, 1998.
- CORKILL, D. D.; LESSER, V. R. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, v. 4, n. 3, p. 15–33, Fall 1983.
- DAML. The DARPA Agent Markup Language. 2001. Disponível em: http://www.daml.org. Acesso em: 13 Feb. 2003.
- DAVID, N. Modelling and Implementing AND and OR-Dependencies in Social Reasoning. Dissertação (Dissertação de Mestrado) Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal, 1998.
- DAVID, N.; SICHMAN, J. S.; COELHO, H. Extending social reasoning to cope with multiple partner coalitions. In: GARIJO, F. J.; BOMAN, M. (Ed.). *Multi-Agent Systems Engineering*. Berlin, DE: Springer-Verlag, 1999. (Lecture Notes in Artificial Intelligence, v. 1647), p. 175–187.
- DAVID, N.; SICHMAN, J. S.; COELHO, H. Agent-based social simulation with coalitions in social reasoning. In: MOSS, S.; DAVIDSSON, P. (Ed.). *Multi-Agent-Based Simulation*. Berlin, DE: Springer-Verlag, 2000. (Lecture Notes in Artificial Intelligence, v. 1979), p. 244–265.
- DECKER, K. S. TÆMS: A framework for environment centered analysis and design of coordination mechanisms. In: O'HARE, G. M. P.; JENNINGS, N. (Ed.). *Foundations of Distributed Artificial Intelligence*. Baffins Lane, UK: John Wiley & Sons Ltd., 1996. p. 429–447.
- DEMAZEAU, Y. From interactions to collective behaviour in agent-based systems. In: *Pre-proceedings of the invited lectures of the 1st European Conference on Cognitive Science*. St. Malo, France: ARC / INRIA, 1995.
- DEMAZEAU, Y.; MÜLLER, J.-P. Decentralized artificial intelligence. In: DEMAZEAU, Y.; MÜLLER, J.-P. (Ed.). *Decentralized A. I.* Amsterdam, NL: Elsevier Science Publishers B. V., 1990. p. 3–13.
- DIGNUM, V.; DIGNUM, F. Modelling agent societies: Co-ordination frameworks and institutions. In: BRAZDIL, P.; JORGE, A. (Ed.). Advances in AI (Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)). Berlin, DE: Springer-Verlag, 2001, (Lecture Notes in Artificial Intelligence, v. 2558). p. 191–204.
- DOYLE, J. Rationality and its role in reasoning. *Computational Intelligence*, v. 8, n. 376–409, 1992.

- DROGOUL, A. De la Simulation Multi-Agents à la Résolution Collective de Problèmes: Une Ètude de l'Èmergence de Structures d'Organisation dans les Systèmes Multi-Agents. Tese (Thèse de Doctorat) Université Paris VI, Paris, France, November 1993.
- DURFEE, E. H. Distributed problem solving and planning. In: WEISS, G. (Ed.). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999. p. 121–164.
- ENGELMORE, R. S.; MORGAN, A. (Ed.). *Blackboard Systems*. Reading, USA: Addison-Wesley, 1988.
- EPSTEIN, J. M.; AXTELL, R. Growing Artificial Societies: Social Science from the Bottom Up. Cambridge, MA, USA: MIT Press, 1996.
- ERCEAU, J.; FERBER, J. L'intelligence artificielle distribuée. *La Recherche*, v. 22, n. 233, p. 750–758x, June 1991.
- ERMAN, L. D. et al. The hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, v. 12, n. 2, p. 213–253, June 1980.
- FARQUHAR, A.; FIKES, R.; RICE, J. The ontolingua server: A tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, v. 46, p. 707–727, 1997.
- FARRENY, H.; GHALLAB, M. *Eléments d'Intelligence Artificielle*. Paris, FR: Hermès, 1987.
- FERBER, J. Les Systèmes Multi-Agents: Vers une Intelligence Collective. Paris, France: InterEditions, 1995.
- FERBER, J.; GASSER, L. *Intelligence Artificielle Distribuée*. 1991. EC2, Avignon, FR. Tutorial Notes of the 11th Conference on Expert Systems and their Applications (Avignon'91).
- FERBER, J.; GUTKNECHT, O. A meta-model for the analysis and design of organizations in multi-agents systems. In: DEMAZEAU, Y. (Ed.). *Proceedings of the 3rd International Conference on Multi-Agent Systems*. Paris, France: IEEE Computer Society Press, 1998. p. 128–135.
- FERGUSON, I. A. Touringmachines: An Architecture for Dynamic, Rational, Mobile Agents. Tese (PhD Thesis) Computer Laboratory, University of Cambridge, Cambridge, UK, October 1992.
- FERREIRA, P.; WAINER, J. Scheduling meetings through multi-agent negotiation. In: MONARD, M. C.; SICHMAN, J. S. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 2000. (Lecture Notes in Artificial Intelligence, v. 1952), p. 126–135.

- FININ, T. et al. KQML as an agent communication language. In: Proceedings of the 3rd International Conference on Information and Knowledge Management. Gaithersburg, MD.: ACM Press, 1994.
- FIPA. The Foundation for Intelligent Physical Agents. 2001. Disponível em: http://www.fipa.org. Acesso em: 13 Feb. 2003.
- FOX, M. S. An organizational view of distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, v. 11, n. 1, p. 70–80, January 1981.
- FOX, M. S. et al. An organizational ontology for enterprise modeling. In: PRIETULA, M. J.; CARLEY, K. M.; GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Cambridge, MA, USA: MIT Press, 1998. p. 131–152.
- GALBRAITH, J. Designing Complex Organizations. Reading, USA: Addison-Wesley, 1973.
- GARCIA, A. C. B.; SICHMAN, J. S. Agentes e sistemas multiagentes. In: REZENDE, S. O. (Ed.). Sistemas Inteligentes: Fundamentos e Aplicações. Barueri, São Paulo, Brasil: Editora Manole Ltda., 2003.
- GASSER, L.; BRAGANZA, C.; HERMAN, N. MACE: A flexible testbed for distributed AI research. In: HUHNS, M. N. (Ed.). *Distributed Artificial Intelligence*. London, UK: Pitman Publishing, 1987. p. 119–152.
- GMYTRASIEWICZ, P. J.; DURFEE, E. H. A representationalist theory of intention. In: BAJCSY, R. (Ed.). *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. Chambery, France: Morgan Kaufmann Publishers, Inc., 1993. v. 1, p. 396–402.
- GROSZ, B.; KRAUS, S. Collaborative plans for complex group action. *Artificial Intelligence*, v. 86, p. 260–357, 1996.
- HANNOUN, M. MOISE: Un modèle organisationnel pour les systèmes multi-agents. Tese (Thèse de Doctorat) Ecole Nationale Supérieure des Mines de Saint-Eienne, Saint-Etienne, France, December 2002.
- HANNOUN, M. et al. MOISE: Un modèle organisationnel pour la conception de systèmes multi-agents. In: Acts des 7èmes Journées Francophones Intelligence Artificielle Distribuée & Systèmes Multi-Agents. Saint-Denis, Ile de la Reunion, France: Herm/'es Science Publications, 1999. p. 105–118.
- HANNOUN, M. et al. MOISE: An organizational model for multi-agent systems. In: MONARD, M. C.; SICHMAN, J. S. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 2000. (Lecture Notes in Artificial Intelligence, v. 1952), p. 152–161.
- HANNOUN, M. et al. Dependence relations between roles in a multi-agent system: Towards the detection of inconsistencies in organization. In: SICHMAN,

- J. S.; CONTE, R.; GILBERT, N. (Ed.). Multi-Agent Systems and Agent-Based Simulation. Berlin, DE: Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v. 1534), p. 169–182.
- HAYES-ROTH, F.; WATERMAN, D.; LENAT, D. B. (Ed.). *Building Expert Systems*. Reading, USA: Addison-Wesley, 1983.
- HEWITT, C. E. Offices are open systems. In: BOND, A. H.; GASSER, L. (Ed.). Readings in Distributed Artificial Intelligence. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1986. p. 312–329.
- HEWITT, C. E. Some requirements for mobile distributed telecomputing architecture. In: CASTELFRANCHI, C.; WERNER, E. (Ed.). *Artificial Social Systems*. Berlin, DE: Springer-Verlag, 1993, (Lecture Notes in Artificial Intelligence, v. 830). p. 259–270.
- HUBERMAN, B. A.; HOGG, T. Communities of Practice: Performance and Evolution. [S.l.], 1994. Technical report.
- HÜBNER, J. F. Um Modelo de Reorganização de Sistemas Multiagentes. Tese (Tese de Doutorado No prelo.) Programa de Pós-Graduação em Engenharia Elétrica, Escola Politécnica da USP, São Paulo, Brasil, 2003.
- HÜBNER, J. F.; SICHMAN, J. S. SACI: Uma ferramenta para implementação e monitoração da comunicação entre agentes. In: USP. *IBERAMIA/SBIA 2000: Open Discussion Track Proceedings*. Atibaia, Brasil, 2000. p. 47–56.
- HÜBNER, J. F.; SICHMAN, J. S. Um modelo de organização para SMA direcionado à reorganização. In: USP. *IBERAMIA/SBIA 2000 Workshop Proceedings*. Atibaia, Brasil, 2000. p. 191–199.
- HÜBNER, J. F.; SICHMAN, J. S.; BECERRA, J. R. Aplicação da arquitetura de objetos distribuídos em sistemas multi-agentes. In: FURB. *Anais do IX Seminário de Computação*. Blumenau, Brasil, 2000. p. 37–47.
- HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional and deontic specification of a mas organization. In: *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems.* Bologna, Italy: New York: ACM,c2002, 2002. p. 501–502.
- HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional and deontic specification of organizations in multiagent systems. In: BITTENCOURT, G.; RAMALHO, G. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 2002. (Lecture Notes in Artificial Intelligence, v. 2507), p. 118–128.
- HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. Spécification structurelle, fonctionnelle et déontique d'organisations dans les sma. In: MATHIEU PHILIPPE ET MÜLLER, J.-P. (Ed.). Actes des 10èmes Journées Francophones Intelligence Artificielle Distribuée & Systèmes Multi-Agents. Lille, France: [s.n.], 2002.

- ITO, M. Uma análise comparativa do fluxo de mensagens entre os modelos da rede contractual (RC*) e coalisões baseadas em dependências (CBD). Dissertação (Dissertação de Mestrado) Programa de Pós-Graduação em Engenharia Elétrica, Escola Politécnica da USP, São Paulo, Brasil, 1999.
- ITO, M.; SICHMAN, J. S. Dependence based coalitions and contract net: a comparative analysis. In: MONARD, M. C.; SICHMAN, J. S. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 2000. (Lecture Notes in Artificial Intelligence, v. 1952), p. 102–111.
- KARAM, M. Description d'Agents vers une Programmation Orientée Agent. Dissertação (Rapport de DEA Informatique) ENSIMAG, INPG, Grenoble, France, 1995.
- KITANO, H. et al. Robocup: The robot world cup initiative. In: *Proceedings* of the *IJCAI'95 Workshop on Entertainment and AI/Alife*. Montreal, Canada: IJCAI Press, 1995.
- KONOLIGE, K. A Deduction Model of Belief. London, UK: Pitman Publishing, 1986.
- KONOLIGE, K.; NILSSON, N. J. Multiple-agent planning systems. In: *Proceedings of the 1st National Conference on Artificial Intelligence*. Stanford, CA: The MIT Press, 1980. p. 138–141.
- LAWLER, E. L. The quadratic assignment problem. *Management Science*, v. 9, p. 586–599, 1963.
- LAWLER, E. L. et al. *The Traveling Salesman Problem*. Baffins Lane, UK: John Wiley & Sons Ltd., 1985.
- LE PAGE, C. et al. CORMAS: A multiagent simulation toolkit to model natural and social dynamics at multiple scales. In: *Proceedings of the Workshop on the Ecology of Scales*. Wageningen, The Netherlands: [s.n.], 2000.
- LEMAÎTRE, C.; EXCELENTE, C. B. Multi-agent organization approach. In: *Proceedings of the 2nd Ibero-American Workshop on DAI/MAS*. Toledo, Spain: [s.n.], 1998. p. 7–16.
- LENAT, D. B. BEINGs: Knowledge as interacting experts. In: BOND, A. H.; GASSER, L. (Ed.). *Readings in Distributed Artificial Intelligence*. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1975. p. 161–168.
- LEVESQUE, H. J.; COHEN, P. R.; NUNES, J. H. T. On acting together. In: *Proceedings of the 8th National Conference on Artificial Intelligence*. Boston, MA: Morgan Kaufmann Publishers, Inc., 1990. p. 94–99.
- LIMA, V. L. S. et al. Uso de sistemas multi-agentes no processamento de linguagem natural: Estudando o problema através do projeto NALAMAS. In: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Anais do 1º Encontro Nacional de Inteligência Artificial. Brasília, Brasíl, 1997.

- LOMBORG, B. Game theory vs multiple agents: The iterated prisoner's dilemma. In: CASTELFRANCHI, C.; WERNER, E. (Ed.). *Artificial Social Systems*. Berlin, DE: Springer-Verlag, 1993, (Lecture Notes in Artificial Intelligence, v. 830). p. 69–93.
- LUGO, G. A. G.; HÜBNER, J. F.; SICHMAN, J. S. Representação e evolução de esquemas sociais em sma: um enfoque funcional. In: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Anais do 3º Encontro Nacional de Inteligência Artificial. Fortaleza, Brasil, 2001.
- LUGO, G. G. et al. Enriching information agents' knowledge by ontology comparison: A case study. In: FRANCISCO J. GARIJO, J. C. R.; TORO, M. (Ed.). Advances in Artificial Intelligence Iberamia 2002. Sevilha: Springer-Verlag, 2002. (Lecture Notes in Artificial Intelligence, v. 2527), p. 546–555.
- MALONE, T. W. What is Coordination Theory. MIT, Boston, USA: MIT Sloan School of Management, 1988.
- MARIETTO, M. d. G. B. et al. Requirements analysis of agent-based simulaton platforms: State of the art and new prospects. In: SICHMAN, J. S.; BOUSQUET, F.; DAVIDSSON, P. (Ed.). *Multi-Agent-Based Simulation II*. Berlin, DE: Springer-Verlag, 2002. (Lecture Notes in Artificial Intelligence, v. 2581).
- MÜLLER, H. J. Negotiation principles. In: O'HARE, G. M. P.; JENNINGS, N. (Ed.). Foundations of Distributed Artificial Intelligence. Baffins Lane, UK: John Wiley & Sons Ltd., 1996. p. 211–230.
- NALAMAS. *Projeto NALAMAS*. 1999. Disponível em: http://www.inf.pucrs.br/~linatural/nalamas/>. Acesso em: 20 Feb. 2003.
- NICOL, J. R.; WILKES, T.; MANOLA, F. A. Obejct orientation in heterogeneous distributed computing systems. *IEEE Computer*, p. 57–67, June 1993.
- NILSSON, N. J. Teleo-reactive programs for agent control. *Journal of Artificial Intelligence*, v. 1, p. 139–158, 1994.
- O'HARE, G. M. P.; JENNINGS, N. (Ed.). Foundations of Distributed Artificial Intelligence. Baffins Lane, UK: John Wiley & Sons Ltd., 1996.
- PAIVA, D. d. S. Ambigüidade categorial em um sistema de processamento de linguagem natural baseado em classes de palavras. In: LIMA, V. L. S.; LEAL, A. L.; SILVA, J. L. T. (Ed.). *Anais do 2o. Workshop do projeto NALAMAS*. Porto Alegre, Brasil: Cópias Todeschini, 1997. p. 34–36.
- PARUNAK, H. V. D. Applications of distributed artificial intelligence in industry. In: O'HARE, G. M. P.; JENNINGS, N. (Ed.). *Foundations of Distributed Artificial Intelligence*. Baffins Lane, UK: John Wiley & Sons Ltd., 1996. p. 139–164.

- PARUNAK, H. V. D. Intelligent agents. In: WEISS, G. (Ed.). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999. p. 377–421.
- PATTISON, H. E.; CORKILL, D. D.; LESSER, V. R. Instantiating description of organizational structures. In: HUHNS, M. N. (Ed.). *Distributed Artificial Intelligence*. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1987. p. 59–96.
- PRIETULA, M. J.; CARLEY, K.; GASSER, L. A computational approach to organizations and organizing. In: PRIETULA, M. J.; CARLEY, K. M.; GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Cambridge, MA, USA: MIT Press, 1998. p. xiii–xix.
- PRUITT, D. G. Negotiation Behaviour. New York, USA: Academic Press, 1981.
- RABAK, C. Otimização do Processo de Inserção Automática de Componentes Eletrônicos Empregando a Técnica de Times Assincronos. Dissertação (Dissertação de Mestrado) Programa de Pós-Graduação em Engenharia Elétrica, Escola Politécnica da USP, São Paulo, Brasil, 1999.
- RABAK, C.; SICHMAN, J. S. Otimização do processo de inserção automática de componentes eletrônicos empregando a técnica de times assíncronos. *Pesquisa Operacional*, v. 21, n. 1, p. 39–59, 2001.
- RAO, A. S.; GEORGEFF, M. P. Modeling rational agents within a BDI-architecture. In: FIKES, R.; SANDEWALL, E. (Ed.). *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1991. p. 473–484.
- RESNICK, M. Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. Cambridge, MA, USA: MIT Press, 1997.
- ROSENSCHEIN, J. S.; ZLOTKIN, G. Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. Cambridge, MA, USA: MIT Press, 1994.
- RUSSEL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach. Englewood Cliffs, NJ, USA: Prentice-Hall International, 1995.
- SACI. SACI: Simple Agent Communication Interface. 2000. Disponível em: http://www.lti.pcs.usp.br/saci. Acesso em: 13 Feb. 2003.
- SEARLE, J. Speach Acts. Cambridge, UK: Cambridge University Press, 1969.
- SHOHAM, Y. Agent oriented programming. *Artificial Intelligence*, v. 60, n. 1, p. 51–92, 1991.

- SICHMAN, J. S. Uma Ferramenta de Monitoração para um Núcleo de Resolução Distribuída de Problemas Orientado a Objetos. Dissertação (Dissertação de Mestrado) Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brasil, June 1991.
- SICHMAN, J. S. Du Raisonnement Social Chez les Agents: Une Approche Fondée sur la Théorie de la Dépendance. Tese (Thèse de Doctorat) Institut National Polytechnique de Grenoble, Grenoble, France, 1995.
- SICHMAN, J. S. O ambiente MASENV e seu uso no projeto NALAMAS. In: LIMA, V. L. S.; LEAL, A. L.; SILVA, J. L. T. (Ed.). *Anais do 2o. Workshop do projeto NALAMAS*. Porto Alegre, Brasil: Cópias Todeschini, 1997. p. 6–8.
- SICHMAN, J. S. On achievable goals and feasible plans in open multi-agent systems. In: *Proceedings of IFAC/IFIC Conference on Management and Control of Production and Logistics (MCPL'97)*. Campinas, Brasil: Elsevier Science Ltd., 1997.
- SICHMAN, J. S. DEPINT: Dependence-based coalition formation in an open multi-agent scenario. *Journal of Artificial Societies and Social Simulation*, v. 2, n. 1, p. http://www.soc.surrey.ac.uk/JASSS/1/2/3.html, 1998.
- SICHMAN, J. S.; CARDOZO, E. A monitoring facility for an object-oriented distributed problem solving kernel. In: *Actes des 1eres. Journées Representations par Objets: Le Point sur la Recherche et les Applications.* La Grande Motte, France: Paris: EC2, 1992. p. 281–290.
- SICHMAN, J. S.; CARDOZO, E. Uma ferramenta de monitoração para um núcleo de resolução distribuída de problemas orientado a objetos. In: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Anais do 19º Seminário Integrado de Hardware e Software. Rio de Janeiro, Brasil, 1992. p. 257–268.
- SICHMAN, J. S.; CONTE, R. On personal and role mental attitudes: a preliminary dependence-based analysis. In: OLIVEIRA, F. M. d. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v. 1515), p. 1–10.
- SICHMAN, J. S.; CONTE, R. Multi-agent dependence by dependence graphs. In: *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems.* Bologna, Italy: ACM Press, 2002. p. 483–492.
- SICHMAN, J. S. et al. A social reasoning mechanism based on dependence networks. In: COHN, T. (Ed.). *Proceedings of the 11th European Conference on Artificial Intelligence*. Amsterdam, The Netherlands: John Wiley & Sons Ltd., 1994. p. 188–192.
- SICHMAN, J. S. et al. A social reasoning mechanism based on dependence networks. In: HUHNS, M.; SINGH, M. (Ed.). *Readings on Agents*. San Mateo, CA, USA: Morgan Kaufmann Publishers, Inc., 1998.

- SICHMAN, J. S.; DEMAZEAU, Y. Exploiting social reasoning to deal with agency level inconsistency. In: LESSER, V. (Ed.). *Proceedings of the 1st International Conference on Multi-Agent Systems*. San Francisco, USA: MIT Press, 1995. p. 352–359.
- SICHMAN, J. S.; DEMAZEAU, Y. Exploiting social reasoning to enhance adaptation in open multi-agent systems. In: WAINER, J.; CARVALHO, A. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 1995. (Lecture Notes in Artificial Intelligence, v. 991), p. 253–263.
- SICHMAN, J. S.; DEMAZEAU, Y. A model for the decision phase of autonomous belief revision in open multi-agent systems. *Journal of the Brazilian Computer Society*, v. 3, n. 1, p. 40–50, Jul 1996.
- SICHMAN, J. S.; DEMAZEAU, Y. On social reasoning in multi-agent systems. Revista Iberoamericana de Inteligencia Artificial, n. 13, p. 68–84, 2001.
- SILVA, J. L. T. d.; ABRAHÃO, P. R. C.; LIMA, V. L. S. d. Integrating morphological, syntactical and semantical aspects through multi-agent cooperation. In: OLIVEIRA, F. M. d. (Ed.). *Advances in AI*. Berlin, DE: Springer-Verlag, 1998. (Lecture Notes in Artificial Intelligence, v. 1515), p. 83–92.
- SIMCOG. The SIMCOG Project. 2002. Disponível em: http://www.lti.pcs.usp.br/SimCog. Acesso em: 13 Feb. 2003.
- SIMON, H. A. Models of Man. New York: Wiley, 1957.
- SMITH, R. G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, v. 29, n. 12, p. 1104–1113, December 1980.
- SMITH, R. G.; RANDALL, D. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, v. 11, n. 1, p. 61–70, January 1981.
- SOUZA, P. d.; TALUKDAR, S. Asynchronous organizations for multi-algorithm problems. In: *Proceedings of ACM Symposium on Applied Computing*. Indianapolis, USA: ACM Press, 1993. p. 286–293.
- STEFIK, M.; BOBROW, D. G. Object-oriented programming: Themes and variations. *AI Magazine*, v. 6, n. 4, p. 40–64, Winter 1983.
- STONE, P.; VELOSO, M. M. Task decomposition and dynamic role assignment for real-time strategic teamwork. In: MÜLLER, J. P.; SINGH, M. P.; RAO, A. S. (Ed.). Intelligent Agents V (Proceedings of the 5th. International Workshop on Agent Theories, Architectures and Languages). Berlin, DE: Springer-Verlag, 1999. v. 1555.
- SUN. Java TM 2 Platform Micro Edition (J2ME TM): Technology for Creating Mobile Devices. Palo Alto, CA: Sun Microsystems, Inc., 2000. White Paper.

SUTTON, R. S.; BARTO, A. G. Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998.

SWARM. The SWARM Development Group. 1999. Disponível em: http://www.swarm.org. Acesso em: 13 Feb. 2003.

TAMBE, M. Towards flexible teamwork. *Journal of Artificial Intelligence*, v. 7, p. 83–124, 1997.

TEAMBOTS. TEAMBOTS. 2000. Disponível em: http://www.teambots.org. Acesso em: 13 Feb. 2003.

TOKORO, M. The society of objects. Tokyo, Japan, December 1993. Technical Report SCSL-TR-93-018.

TRAVERS, J.; MILGRAM, S. An experimental study of the small world problem. *Sociometry*, v. 32, n. 4, p. pp.425–443, 1969.

WEISS, G. (Ed.). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1999.

WOOLDRIDGE, M. Intelligent agents. In: WEISS, G. (Ed.). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999. p. 27–77.

WOOLDRIDGE, M. An Introduction to MultiAgent Systems. Baffins Lane, UK: John Wiley & Sons Ltd., 2002.

WOOLDRIDGE, M.; JENNINGS, N. R. Towards a theory of cooperative problem solving. In: DEMAZEAU, Y.; MÜLLER, J.-P.; PERRAM, J. (Ed.). Pre-proceedings of the 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Odense, Denmark: [s.n.], 1994. p. 15–26.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, Amsterdam, The Netherlands, v. 10, n. 2, p. 115–152, 1995.

WOOLDRIDGE, M. J.; JENNINGS, N. R. Agent theories, architectures and languages: A survey. In: WOOLDRIDGE, M. J.; JENNINGS, N. R. (Ed.). *Intelligent Agents*. Berlin, DE: Springer-Verlag, 1995, (Lecture Notes in Artificial Intelligence, v. 890). p. 1–39.

XML. The eXtensible Markup Language. 2001. Disponível em: http://www.xml.org. Acesso em: 13 Feb. 2003.

Apêndice I – Exemplo de código de agente no ambiente \mathcal{M} asenv

O trecho de código mostrado a seguir, apresentado originalmente em (CARDO-ZO; SICHMAN; DEMAZEAU, 1993), é um exemplo de utilização do ambiente \mathcal{M} ASENV. Tal trecho representa o início da codificação de um agente, quando este anuncia sua entrada na sociedade.

```
#include "agent.h"
extern Shared_class *Mbox, *Message;
Active_object *mailbox; // my mailbox
Active_object *message; // generic message
Local_object *obj; // a local object
                // message's content
char content;
int agent_id;  // this agent ID
// initialize agent and create its mailbox
agent_id = Initialize_agent();
mailbox = Mbox->Make_instance();
// fill the message's contents
content = "build_2D_description gaussian ...";
// buid object containing the message's slots
obj = new Local_object("Message",
     &Slot("Info_type", INFORM),
```

```
&Slot("Ellocution_force", COMMANDING"),
      &Slot("Info_type", (ROLE | BASIC_ACTION)),
      &Slot("Format", "%s %s"),
      &Slot("Content", content),
      &Slot("Message_id", Message_counter++),
      &Slot("Receiver", BROADCAST),
      &Slot("Sender", agent_id));
// build a message with slots above
message = Message->Make_instance(obj, NULL);
// send it calling the method "Send"
Mbox->CallMethod("Send", message);
// wait for other agent's reactions
message = Mbox->CallMethod("Receive");
if(message) // process message
  {
  }
```

As primeiras duas facetas da mensagem especificam o ato de fala utilizado na mensagem: no caso do exemplo, trata-se de uma informação (INFORM), já que não houve uma solicitação anterior para enviá-la, e que deve ser registrada pelos outros agentes (COMMANDING). As facetas Message_class e Info_type especificam que a mensagem contém o papel (ROLE) e as ações básicas (BASIC_ACTION) do novo agente, e que portanto devem ser consideradas como parte de sua descrição externa. As informações dependentes de domínio contidas nas facetas Format e Content são típicas de uma aplicação em visão computacional. As demais facetas referem-se à identificação da mensagem (utilizada quando o agente encontra-se engajado em mais de uma interação simultânea), o emissor e receptor da mensagem.

O ambiente atualiza dinamicamente duas facetas: Number_of_readers e Sender_state. A primeira indica o número de agentes que ainda não leram a mesagem. Já a segunda mantém o estado do emissor da mensagem, indicando ao receptor, por exemplo, se o agente emissor ainda se encontra na sociedade. Esta última informação é necessária, já que o ambiente se propõe a servir de base para o desenvolvimento de sistema multagentes abertos, e neste caso um agente pode abandonar a sociedade antes que os outros tenham tido a oportunidade de ler as mensagens por ele enviadas.

O objeto ativo que contém a mensagem é automaticamente destruído depois que todos os agentes receptores da mensagem já a tiverem lido.

Apêndice II – Exemplo de código de agente no ambiente \mathcal{S} aci

O trecho de código mostrado a seguir, apresentado originalmente em (HÜBNER; SICHMAN, 2000a), é um exemplo de utilização do ambiente \mathcal{S} ACI. Tal trecho implementa um agente que tem a habilidade de somar e disponibiliza esta funcionalidade à sua sociedade.

```
import saci.*;
public class PlusServer extends Agent {
 public static void main(String[] args) {
   Agent a = new PlusServer();
   Config c = new Config();
   c.set("facilitator.host", "nantes");
   if (a.enterSoc("Ag2", c)) {
     a.initAg();
     a.run();
 } }
 public void initAg() {
   try {
     mbox.advertise("ask-one", null, "math", "X+Y");
 } } catch (Exception e) { ... }
 public void run() {
   while (true) {
     Message m = mbox.polling();
     if (m.get("performative").equals("ask-one")) {
       Message r = new Message("(tell)");
       r.put("receiver", m.get("sender"));
```

```
r.put("in-reply-to", m.get("reply-with"));
    r.put("content", sum(m.get("content")));
    mbox.sendMsg(r);
} }
String sum(Object formula) { ... }
}
```

O ciclo de vida do agente está codificado nas linhas 7 a 9: o agente entra na sociedade sugerindo uma identificação Ag2, anuncia suas habilidades (através do método initAg) e inicia sua execução (recebimento e envio de mensagens no método run). O método initAg, na linha 11, envia um advertise ao facilitador, indicando que o agente pode receber mensagens com a performativa ask-one, com conteúdo escrito em qualquer linguagem — indicado pela cadeia null —, na ontologia math e com formato X+Y. As funcionalidades do Saci são utilizadas através do objeto mbox. Por exemplo, na linha 13 é feito o anúncio nas páginas amarelas do facilitador, na linha 17 o agente espera até que uma mensagem chegue em sua caixa postal, nas linhas 19 a 22 uma mensagem KQML é construída, e na linha 23 esta mensagem é enviada assincronamente.

Um agente que solicita o serviço do agente desenvolvido acima poderia ter o seguinte método run:

```
" answers is " + r.get("content"));
}
} catch (Exception e) { ... }
}
```

O método consultYP (linha 4) encapsula a consulta ao facilitador pelos agentes que tem a capacidade de somar. Para cada um destes agentes é enviada uma mensagem (composta nas linhas 10 a 12) solicitando a realização de uma soma (linha 13). O método utilizado para esta solicitação (ask(m)) envia uma mensagem de performativa ask e espera a resposta para esta mensagem.

Apêndice III – Exemplo de sessão de utilização do sistema \mathcal{D} epint

Os resultados mostrados a seguir, apresentados originalmente em (SICHMAN, 1998), foram obtidos de uma sessão de utilização do sistema $\mathcal{D}_{\text{EPINT}}$, que ilustram um exemplo de formação de coalizão entre agentes, num domínio de laboratórios de pesquisa acadêmicos.

Seja então uma sociedade composta por 5 (cinco) agentes ag_5 , ag_6 , ag_7 , ag_8 e ag_9 . O agente ag_5 tem 3 (três) objetivos a alcançar, conforme o conteúdo de sua descrição externa, mostrada na tabela III.1.

Como o objetivo mais importante (write_ss_mas_paper) de ag₅ não é atingível, ele escolhe um plano para tentar atingir seu segundo objetivo mais importante (write_mas_paper). Sua situação de objetivo é DEP, pois ele não sabe realizar uma ação (process_latex) necessária para atingí-lo. Sua situação de dependência para este objetivo em relação aos agentes ag₆, ag₇, e ag₈ é respectivamente UD, UD e LBMD. Consequentemente, o agente ag₅ decide então enviar ao agente ag₈, cuja situação de dependência é mais encorajadora, uma proposta de coalizão. Nesta proposta, ele se oferece para realizar uma outra ação (write_mas_section), para que juntos possam atingir o objetivo comum (write_mas_paper). A saída de processamento do sistema, que termina com o envio da proposta de coalizão, conforme o protocolo de comunicação descrito na seção 8.2.5, é apresentada abaixo:

```
===== Initial state ...

Do you want the agent to be active in this cycle? (y/n): y

===== Reasoning about goals ...

My dependence network is:
```

Descrição Externa			
Agente	Objetivos	Planos	Ações
ag_5	write_mas_paper	$write_mas_paper() := \\ write_mas_section(), \\ process_latex().$	write_mas_section
	$write_ss_mas_paper$	$write_ss_mas_paper() := \\ write_ss_section(), \\ write_mas_section(), \\ process_latex().$	$analyse_mas_paper$
	review_oop_paper	$review_oop_paper() := analyse_oop_paper().$ $review_mas_paper := analyse_mas_paper().$	- analyse_oop_paper
ag_6	$write_tel_paper$	$write_tel_paper() := \\ write_tel_section(), \\ process_latex().$	$write_tel_section$
	$review_sig_paper$	$review_sig_paper() := \\ analyse_sig_paper().$	$analyse_tel_paper$
	review_se_paper	$review_se_paper() := \\ analyse_se_paper().$	process_latex
		$review_tel_paper() := \\ analyse_tel_paper().$	process_iatex
ag_7	$write_sig_paper$	$write_sig_paper() := \\ write_sig_section, \\ process_latex().$	$write_sig_section$
	$review_tel_paper$	$review_tel_paper() := \\ analyse_tel_paper().$	$analyse_sig_paper$
	review_se_paper	$review_se_paper() := \\ analyse_se_paper().$	process_latex
		$review_sig_paper() := \\ analyse_sig_paper().$	-
ag_8	$write_mas_paper$		$process_latex$
ag_9	$write_ss_mas_paper$		

Tabela III.1: Exemplo de descrição externa no sistema \mathcal{D} EPINT (SICHMAN, 1998).

```
ag5
 <ag5>
----- write_mas_paper (20)
        |----- write_mas_paper:=write_mas_section(),
                                 process_latex().
                |---- process_latex
                         |---- ag6
                                  |----
                                  l ag8
        | review_oop_paper (10)
        |----- review_oop_paper:=analyse_oop_paper().
                |---- A-AUTONOMOUS
                         |-----
         write_ss_mas_paper (30)
        |----- write_ss_mas_paper:=write_ss_section(),
                                    write_mas_section(),
                                    process_latex().
                 |---- write_ss_section
                         | ******* UNKNOWN
                                 |----
                         | process_latex
                         |***** ag6
                                  |----
                                  l ag7
                                  |----
                                  l ag8
                                  |----
My current list of possible goals is :
    write_mas_paper(20) achievable
    review_oop_paper(10) achievable
    write_ss_mas_paper(30) non achievable
```

```
===== Deciding about goals ...
The goal selected is: write_mas_paper (20)
===== Reasoning about plans ...
My dependence network is:
  ag5
 <ag5>
---- write_mas_paper (20)
        |----- write_mas_paper:=write_mas_section(),
                                    process_latex().
                  |----- process_latex
                           |---- ag6
                                    |-----
                                     |-----
                                     l ag8
My current list of possible plans is:
    write_mas_paper:=write_mas_section(), process_latex().(20)
                                                   feasible
==== Deciding about plans ...
The plan selected is :
    write_mas_paper:=write_mas_section(), process_latex(). (20)
===== Reasoning about partners ...
My dependence network is:
 ag5
 <ag5>
----- write_mas_paper (20)
        |----- write_mas_paper:=write_mas_section(),
                                    process_latex().
                  |----- process_latex
```

```
|---- ag6
                                      |----
                                      l ag7
                                      |----
                                      l ag8
My goal situation is DEP
My needed action is process_latex
My current list of partners is :
     (ag6) UD NONE NONE
     (ag7) UD NONE NONE
     (ag8) LBMD write_mas_paper write_mas_section
===== Deciding about partners ...
The partner selected is:
               (ag8) LBMD write_mas_paper write_mas_section
==== Sending a message ...
---> Etape : conv1 DEPINTPROPOSITION init receive "5.5.92"
---> Destinataire : you
---> Type : request
---> Ressource : matter=dec put moi proposal
There is only one possible transition in the protocol
==== Trying to receive a message ...
```

Ao receber a proposta de coalizão, o agente ag_8 calcula sua situação de objetivo para o objetivo oferecido ($write_mas_paper$) e verifica que esta é NP. Assim, como não tem nenhum plano em seu repertório que permita atingir este objetivo, resolve aceitar a proposta de coalizão. A saída de processamento do sistema, que termina com o envio da mensagem de aceitação da proposta de coalizão, conforme o protocolo de comunicação descrito na seção 8.2.5, é a seguinte:

```
==== Initial state \dots Do you want the agent to be active in this cycle? (y/n): n
```

```
Do you want the agent to leave the society? (y/n): n
==== Inferring properties about other agents ...
Do you want the agent to infer in this cycle? (y/n): n
==== Perceiving properties of other agents ...
Do you want the agent to perceive in this cycle? (y/n): n
===== Trying to receive a message ...
The message received is:
  ( PROPOSAL < ag5 polaris.imag.fr 13892 >
  (write_mas_paper process_latex LBMD
   write_mas_paper write_mas_section )
==== Reasoning about messages ...
I have received a proposal of coalition:
  ( PROPOSAL < ag5 polaris.imag.fr 13892 >
  (write_mas_paper process_latex LBMD
   write_mas_paper write_mas_section )
My dependence network is:
  ag8
 <ag8>
---- write_mas_paper (10)
         |---- NO-PLANS
                   |----
My goal situation is NP
===== Deciding about proposals ...
I will accept the proposal, because I do not have a plan for this goal
==== Sending a message ...
Si error
---> Etape : conv1 DEPINTPROPOSITION init end "5.5.92"
```

```
---> Destinataire : you
---> Type : answer
---> Ressource : matter=obs,put,moi,revision,

Si !error&&!best_option
---> Etape : conv1 DEPINTPROPOSITION init end "5.5.92"
---> Destinataire : you
---> Type : answer
---> Ressource : matter=dec,put,moi,refusal,

Si !error&&best_option
---> Etape : conv1 DEPINTPROPOSITION init end "5.5.92"
---> Destinataire : you
---> Type : answer
---> Ressource : matter=dec,put,moi,acceptance,

The transition chosen is !error&&best_option
```

Finalmente, o agente ag_5 recebe a resposta do agente ag_8 e a coalizão é formada, conforme mostrado a seguir:

```
===== Trying to receive a message ...
The message received is:
          ( ACCEPTANCE < ag8 polaris.imag.fr 13895 > )
===== Reasoning about messages ...
*** The partner has accepted to form a coalition ***
```

Apêndice IV – Exemplo de sessão de utilização do sistema \mathcal{D} epint $^+$

Os resultados mostrados a seguir, apresentados originalmente em (DAVID; SICH-MAN; COELHO, 2000), foram obtidos de uma sessão de utilização do sistema $\mathcal{D}_{\text{EPINT}^+}$, que ilustram um exemplo de tentativa de formação de coalizão entre agentes, num domínio de aviação comercial.

Seja então uma sociedade composta por 3 (três) agentes Af, Tp e Au. Inicialmente, a saída do processamento do sistema mostra a apresentação dos agentes na sociedade:

```
Paris/Sydney:= Paris/Lisbon, Lisbon/Macau, Macau/HongK, HongK/Sydney.
Rome/Boston:= Rome/Paris, Paris/NewY, NewY/Boston.
Rome/Marseille:= Rome/Paris, Paris/Toulouse, Toulouse/Marseille.
==== I have received the following messages of introduction:
Identity:
<Tp tp.north.com 7352>
Goals:
Lisbon/Moscow(300)
Actions:
Lisbon/Paris(26); Lisbon/Macau(156); Macau/HongK(2)
Plans:
Lisbon/Moscow:= Lisbon/Paris, Paris/Moscow.
Identity:
<Au au.anywhere.com 7366>
Goals:
Sydney/SaoPaulo (45)
Actions:
Sydney/BuenosAires(147); Sydney/Pretoria(156); HongK/Sydney(100)
Plans:
Sydney/SaoPaulo:= Sydney/BuenosAires, BuenosAires/SaoPaulo.
```

O agente Af tem 4 (quatro) objetivos a alcançar, e escolhe o objetivo «Paris/Sydney», que é o seu único objetivo atingível. Sua situação de objetivo é DEP, pois ele não sabe realizar algumas das ações necessárias para atingir esse objetivo. A continuação do processamento do sistema, que mostra a fase de raciocínio e decisão sobre objetivos e planos é a seguinte:

```
======= Reasoning about goals ...
My dependence network is:
  <Af>
-- Paris/Sydney (120) (achievable)
```

```
|---- Paris/Sydney:= Paris/London, London/HongK, HongK/Sydney.
      |---- London/HongK (NA)
                                                 (NFeasible)
               | ******** UNKNOWN
                | HongK/Sydney (EC:100.0)
                |****** <Au 7366> (100.0)
  (\dots)
      |Paris/Sydney:= Paris/Lisbon, Lisbon/Macau, Macau/HongK,
      |----- Lisbon/Macau (EC:156.0)
                                                 HongK/Sydney. (EC:284)
                |******* <Tp 7352> (156.0)
                | Macau/HongK (EC:2.0)
                |****** <Tp 7352> (2.0)
                | HongK/Sydney (EC:100.0)
                |****** <Au 7366> (100.0)
  (...)
  | Rome/Marseille (33) (not achievable)
  |---- Rome/Marseille:= Rome/Paris, Paris/Toulouse, Toulouse/Marseille.
       |----- Rome/Paris (NA)
                                                            (NFeasible)
                |---- UNKNOWN
                          |-----
====== Deciding about goals ...
The selected goal is: Paris/Sydney (120)
======= Reasoning about plans ...
(...)
====== Deciding about plans ...
The selected plan is:
  Paris/Sydney:=
Paris/Lisbon, Lisbon/Macau, Macau/HongK, HongK/Sydney. (EC:284)
```

O agente Af escolhe um plano composto pelas ações « Paris/Lisbon », « Lisbon/Macau », « Macau/HongK » e « HongK/Sydney », dado que todos os outros planos para atingir esse objetivo não são factíveis. Este caso representa uma E-dependência, conforme discutido na seção 5.1.7, já que o agente Af necessita que outros agentes desempenhem três das quatro ações citadas acima. Nomeadamente, o agente Af depende do agente Tp para as ações « Lisbon/Macau » e « Macau/HongK » e do agente Au para a ação « HongK/Sydney ».

A fase da escolha de parceiros para a tentativa de formação de coalizões é composta por alguns procedimentos adicionais introduzidos na extensão do modelo $S_{\rm RM}$, apresentados na seção 5.1.12. Tal extensão introduz um modelo de raciocínio sobre a escolha das melhores propostas de coalizão entre várias alternativas possíveis como mostrado a seguir:

```
======= Reasoning about partners ...
My needed actions are:
<Lisbon/Macau>, <Macau/HongK>, <HongK/Sydney>
My possible partners, offered goals, plans and actions
   for each needed action are:
Lisbon/Macau and Macau/HongK
|---- <Tp tp.north.com 7352>
    / d-sit: UD / d-strength: 5.4 / d-a-cost: 158.0
    |---- Lisbon/Moscow (300) (RBID)
        |---- Lisbon/Moscow:= Lisbon/Paris, Paris/Moscow.
            |---- Paris/Moscow (52)
                                      (Feasible NLSource)
                  |----
HongK/Sydney
|---- <Au au.anywhere.com 7366>
    / d-sit: UD / d-strength: 0.0 / d-a-cost: 100.0
    |---- no offered goals
```

```
====== Deciding about partners ...
Partner choices criteria: d-sit > d-strength > action_cost
The selected partner(s) and proposal(s) are :
Needed actions: Lisbon/Macau and Macau/HongK
| Partner: <Tp tp.north.com>
| Offered goal: <Lisbon/Moscow>
| Offered action: <Paris/Moscow>
| Needed action: HongK/Sydney
| Partner: <Au au.anywhere.com>
| Offered goal: NONE
| Offered action: NONE
====== Sending and receiving messages ...
Sending proposals of coalition to
<Tp tp.north.com 7352> ...
<Au au.anywhere.com 7366> ...
The messages received are:
(Acceptance <Tp 7352>), Refusal <Au 7366>)
My new list of possible partners, offered goals, plans and actions is:
HongK/Sydney
|---- no possible partners (empty list)
Informing agent <Tp tp.north.com 7352> that the proposal of coalition
was canceled ...
```

====== Reasoning about goals ...

The selected goal is no longer achievable.

As redes de dependência identificam os parceiros possíveis para cada ação constituinte da E-dependência, assim como a correspondente situação de dependência. Adicionalmente, identificam os objetivos, planos e ações que o agente proponente Af está disposto a contribuir em troca, ou seja, os objetivos, planos e ações para os quais os possíveis parceiros Tp e Au dependem do agente proponente Af. Em ambos os casos observa-se que existe uma UD entre o agente Af e os agentes Tp e Au. Verifica-se também que em troca das ações « Lisbon/Macau » e « Macau/HonK » que o agente Tp sabe realizar, o agente Af está disposto a desempenhar a ação « Paris/Moscow » que o agente Tp não sabe realizar, contribuindo assim para que este atinja o seu objetivo « Lisbon/Moscow ». Esta inferência é possível porque o agente Af utilizou os planos do agente Tp para raciocinar sobre as propostas a lhe oferecer.

Infelizmente para o agente Af, embora ele dependa do agente Au para a ação « HongK/Sydney», o agente Au não depende de Af para quaisquer de suas ações, não se vislumbrando qualquer proposta que o agente Af possa oferecer a Au. Esta diferença ilustra a utilidade em algumas situações de iniciar o raciocínio a partir dos planos em que o possível parceiro acredita, tal como referido na seção 5.1.12. Embora o agente Af calcule para o seu objetivo a mesma UD em relação aos agentes Tp e Au, é patente que o poder de barganha do agente Af com o agente Au será menor do que com o agente Tp.

Desta forma, depois do agente Af enviar as propostas de coalizão para os agentes Tp e Au, verifica-se que o agente Tp aceita a proposta, mas o agente Au a recusa. Esta recusa do agente Au é justificada pelo princípio da nãobenevolência, discutido na seção 8.2.2, pois ele não tem nada a ganhar em aceitar a participação na coalizão com o agente Af. Este exemplo ilustra também o protocolo para o estabelecimento de coalizões com múltiplos parceiros. Como a tentativa de coalizão falhou devido à recusa do agente Au, o agente Af informa o agente Tp que a tentativa de coalizão foi cancelada.