

Jomi Fred Hübner

Um Modelo de Reorganização de Sistemas Multiagentes

Texto apresentado à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Doutor em Enge-
nharia Elétrica.

São Paulo
2003

Jomi Fred Hübner

Um Modelo de Reorganização de Sistemas Multiagentes

Texto apresentado à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Doutor em Enge-
nharia Elétrica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Jaime Simão Sichman

São Paulo
2003

Ficha Catalográfica

Hübner, Jomi Fred

Um Modelo de Reorganização de Sistemas Multiagentes. São Paulo, 2003. 224 p.

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

1. Sistemas Multiagentes. 2. Reorganização de Sistemas Multiagentes. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. Título.

a *Ilze, Morgana e Thales*

“Quando se começa a indagar, nada resiste”

(Simone Beauvoir)

Agradecimentos

Esta tese certamente não é resultado apenas de um esforço individual, mas é resultado de uma ampla rede de colaboração e apoio. Mesmo sabendo que não será possível nomear aqui todas, gostaria de mencionar e agradecer algumas pessoas e instituições que colaboraram com este trabalho.

Inicialmente gostaria de agradecer ao Prof. Jaime Simão Sichman que aceitou orientar este trabalho no decorrer de quatro anos. Além das diretrizes gerais sobre como encaminhar a pesquisa e de indicar os melhores momentos para “devanear” e “focalizar”, gostaria de agradecer particularmente a dedicação (e os fins de semana) com que revisou o texto desta tese e artigos relacionados. Também gostaria de agradecer o esforço do Prof. Jaime em oportunizar minha inclusão na comunidade acadêmica de IA, particularmente no grupo de pesquisa em SMA da École Nationale Supérieure des Mines de Saint-Etienne onde fiz um estágio de seis meses com o Prof. Olivier Boissier.

Ao Prof. Olivier Boissier, e à sua família, tenho uma gratidão especial por receber a mim, e à minha família, em sua casa. Por dar todo o suporte à nossa instalação em Saint Etienne e por nos proporcionar momentos de alegria em sua companhia nos muitos fins de semana que passamos juntos. Além destas alegrias, também agradeço ao Olivier pelas valiosas contribuições ao modelo $\mathcal{M}OISE^+$ e ao estudo de caso E-Alliance (que foram concebidos e implementados no estágio).

Este trabalho não teria se realizado sem o apoio financeiro de algumas instituições. Agradeço, portanto, à CAPES pela bolsa de doutorado, ao CNPq pela bolsa para o estágio e à FURB pela manutenção do salário durante os três anos de afastamento, particularmente aos colegas do Departamento de Sistemas e Computação que se empenham na capacitação do corpo docente. Sou grato também à comunidade de software livre. Todo o desenvolvimento da tese foi realizado com este tipo de software, razão pela qual os sistemas resultantes dessa pesquisa também são abertos.

Aos amigos e amigas do Laboratório de Técnicas Inteligentes: Alexandre, Anna (e seu apoio no uso de aprendizado por reforço), Carlota, Bianchi, Fabrício, Luis, Nuno, Gomi, Graça, Gustavo (pelas inúmeras conversas produtivas, e pelas improdutivas também) e Rafael. Ao amigo de longa data, Rafael Heitor Bordini, pelo incentivo e contribuições à tese e pela “iniciação” ao Linux e ao L^AT_EX (duas ferramentas valiosas para o desenvolvimento de uma pesquisa).

Por fim, agradeço profundamente a minha família (Ilze, Morgana, Thales, Paula, Siegfried *in memoriam*, Jean, Adriana e Dagna) pelo apoio e compreensão durante todo o processo de desenvolvimento desta tese. Em especial, agradeço a Ilze por estar ao meu lado todo este tempo e a ela dedico este trabalho.

Resumo

Esta tese propõe um modelo de adaptação organizacional em Sistemas Multiagentes (SMA) composto por um modelo de organização e um processo de reorganização. O modelo organizacional, chamado de \mathcal{MOISE}^+ , considera três dimensões da organização de um SMA: a estrutura (grupos, papéis e ligações), o funcionamento (planos globais, metas e missões) e as obrigações e permissões dos agentes. A principal característica do modelo \mathcal{MOISE}^+ é facilitar a mudança organizacional em uma das dimensões sem comprometer as outras. É possível, por exemplo, alterar a estrutura da organização sem alterar o seu funcionamento e vice-versa. O processo de reorganização permite que os próprios agentes realizem a mudança através de quatro etapas: monitoração da organização corrente, projetos de planos de mudança, seleção dos planos e implementação do plano selecionado. A principal característica deste processo é a existência explícita da organização responsável pela reorganização (descrita na notação do modelo \mathcal{MOISE}^+) permitindo a *abertura* do processo. Estando a representação deste processo disponível aos agentes, estes podem participar da reorganização, bastando que compreendam a notação do modelo. Este processo foi implementado e avaliado em dois estudos de caso, empresas virtuais e futebol de robôs, nos quais as etapas de seleção e implementação foram enfatizadas, já que existem poucos trabalhos que abordam estas duas etapas do processo de reorganização.

Abstract

This thesis proposes a reorganization model for Multiagent Systems (MAS). This reorganization model is composed by both an organizational model and a reorganization process. The organization model, called \mathcal{MOISE}^+ , is based on three dimensions: the organizational structure (composed by roles, groups, links, etc.), the organizational functioning (composed by global plans, missions, goals, etc.), and the agents' obligations and permissions. The \mathcal{MOISE}^+ model main feature is the independence among these dimensions. It is therefore possible to change the organizational structure without any change in the functioning. The reorganization process enables the agents themselves to perform the reorganization along four phases: monitoring (when to reorganize), design (ways of building a new organization), selection (how to choose a new organization), and implementation (how to change the current running organization). The reorganization process is explicitly described in \mathcal{MOISE}^+ notation and the agents have access to it. This property enables any agent to participate in the reorganization when he understands \mathcal{MOISE}^+ notation. The reorganization process is thus *open* since many agents can joint it. This process was implemented and evaluated in two domains: virtual enterprises and robot soccer. The experiments have focused mainly on the selection and implementation phases since the research on those phases is not very common in the literature.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

Convenções e Lista de Símbolos

1	Introdução	1
1.1	Sistemas Multiagentes	1
1.1.1	Agentes	3
1.1.2	Organização dos agentes	7
1.2	Objetivos	9
1.3	Organização do texto	9
I	Organização	12
2	Organização em Sistemas Multiagentes	13
2.1	Introdução	14
2.1.1	Organização do tipo AR	15
2.1.2	Organização do tipo AC	17
2.1.3	Organização do tipo OR	17
2.1.4	Organização do tipo OC	18

2.1.5	Autonomia organizacional	19
2.2	Definições centradas na organização	20
2.2.1	Modelo organizacional AALAADIN	21
2.2.2	Modelo organizacional de Pattison, Corkill e Lesser	21
2.2.3	Modelo organizacional TOVE	22
2.2.4	Modelo organizacional TÆMS	24
2.2.5	Modelo organizacional MOISE	27
2.2.6	Organização e AOSE	29
2.3	Conclusões	32
3	Modelo <i>Moise</i>⁺	36
3.1	Introdução	36
3.2	Especificação estrutural	39
3.2.1	Nível individual: papéis	39
3.2.2	Nível social: ligações e compatibilidades	41
3.2.3	Nível coletivo: grupos	43
3.3	Especificação funcional	55
3.3.1	Metas globais	56
3.3.2	Nível individual: missões	57
3.3.3	Nível coletivo: esquema social	57
3.3.4	Preferência entre missões	64
3.4	Especificação deôntica	67
3.5	Entidade organizacional	71
3.6	Comparação com outros modelos organizacionais	77
3.7	Conclusões	79

4	Proposta de arquitetura para o modelo $\mathcal{M}oise^+$	84
4.1	Visão geral	85
4.2	Dinâmica da entidade organizacional	86
4.2.1	Criação da entidade organizacional	87
4.2.2	Eventos relacionados aos grupos	88
4.2.3	Eventos relacionados aos esquemas	92
4.2.4	Eventos relacionados aos agentes	96
4.3	Componente OrgBox	100
4.4	Exemplo de utilização da arquitetura	103
4.5	Conclusões	109
II	Reorganização	111
5	Reorganização de Sistemas Multiagentes	112
5.1	Introdução	112
5.2	Monitoração	117
5.3	Projeto	119
5.4	Seleção e Implementação	121
5.5	Conclusões	121
6	Reorganização utilizando o modelo $\mathcal{M}oise^+$	124
6.1	Introdução	124
6.2	Estrutura de reorganização	125
6.3	Monitoração	128
6.4	Projeto	132
6.5	Seleção	135

6.5.1	Seleção por votação	136
6.5.2	Seleção por aprendizado	142
6.5.3	Comparação entre as duas propostas de seleção	147
6.6	Implementação	148
6.7	Conclusões	152

III Estudos de caso 156

7 E-Alliance 157

7.1	Introdução	157
7.2	Primeira reorganização do grupo E-Alliance	160
7.2.1	Monitoração	160
7.2.2	Projeto	162
7.2.3	Seleção e Implementação	165
7.3	Segunda reorganização do grupo E-Alliance	168
7.3.1	Monitoração	169
7.3.2	Projeto	170
7.3.3	Seleção	170
7.4	Conclusões	174

8 JOJTEAM 176

8.1	Introdução	176
8.2	Desenvolvimento do JOJTEAM	177
8.2.1	Organização do time	178
8.2.2	Funcionamento dos agentes	180
8.3	Configuração de reorganização com aprendizado	182

8.3.1	Monitoração	182
8.3.2	Projeto	183
8.3.3	Seleção	187
8.3.4	Resultados	190
8.4	Conclusões	194
9	Conclusões	196
9.1	Contribuições	196
9.2	Trabalhos futuros	198
	Anexos	201
A	SACI	201
A.1	Entrada e saída de agentes	202
A.2	Envio e recebimento de mensagens	202
A.3	Anúncio de habilidades	202
A.4	Programação dos agentes com o SACI	204
B	Exemplo de especificação organizacional no formato XML	207
B.1	Exemplo da escola	207
B.2	Especificação do JOJTEAM	210
	Referências Bibliográficas	214
	Índice Remissivo	223

Lista de Figuras

1.1	Desenvolvimento ideal de Sistemas Multiagentes.	4
1.2	Duas arquiteturas clássicas de agentes.	6
1.3	Dependência entre os capítulos.	10
2.1	Pontos de vista para organização de um SMA.	14
2.2	Tipos de organização.	16
2.3	Modelo AALAADIN de organização segundo Ferber e Gutknecht (1998).	22
2.4	Taxonomia organizacional proposta por Fox et al. (1998).	24
2.5	Visão objetiva de uma tarefa no modelo TÆMS (DECKER, 1998).	25
2.6	Visão subjetiva de uma tarefa no modelo TÆMS (DECKER, 1998).	26
2.7	Descrição de uma missão no modelo MOISE (HANNOUN et al., 2000)	28
2.8	Entidade organizacional no modelo MOISE (HANNOUN et al., 2000).	30
2.9	Relação entre os modelos de especificação da metodologia GAIA (WOOLDRIDGE; JENNINGS; KINNY, 1999).	31
3.1	Contribuição da organização para a finalidade do SMA.	38
3.2	Exemplo de Especificação Estrutural com o modelo MOISE ⁺	49
3.3	Exemplo de EE para seleção de candidatos.	54
3.4	Outra possibilidade para EE de seleção de candidatos.	54
3.5	Exemplo de esquema social para ingresso em um curso de pós- graduação.	58
3.6	Visão simplificada dos constituintes de um SMA segundo o modelo MOISE ⁺	71

3.7	Resumo do modelo proposto segundo a visão organizacional adotada.	83
4.1	Camadas e componentes da arquitetura proposta.	85
4.2	Dependência para remoção de um componente da EnO.	90
4.3	Funções de verificação do estado de uma meta.	95
4.4	Algoritmo que determina as obrigações de um agente.	102
4.5	Algoritmo que determina as metas possíveis para um agente.	103
4.6	Especificação organizacional da escola.	104
4.7	Especificação dos grupos da escola.	105
4.8	Estado inicial do esquema de realização de uma prova.	106
4.9	Estado do esquema de realização de uma prova após os compromettimentos dos agentes.	108
4.10	Estado do esquema de realização de uma prova após a realização da meta <i>preparar prova</i> .	109
4.11	Estado final do esquema de realização de uma prova.	110
5.1	Caracterização de alguns tipos de falha organizacional.	118
5.2	Arquitetura de reorganização proposta por Horling, Benyo e Lesser (2001).	120
5.3	Visão geral de um SMA com capacidade de se auto-reorganizar.	122
6.1	Grupo de reorganização.	126
6.2	Esquema de reorganização.	128
6.3	Grupo de seleção com um Monitor.	131
6.4	Agente no contexto de aprendizado por reforço.	143
6.5	Algoritmo Q-Learning	146
6.6	Dependência para remoção dos componentes da EO.	151
6.7	Resumo do modelo proposto de reorganização.	153
7.1	Primeira especificação organizacional da Aliança.	159

7.2	Primeira entidade organizacional da Aliança.	159
7.3	Monitoração da Aliança.	161
7.4	Projeto de uma nova EO para a Aliança.	162
7.5	Proposta de alteração enviada pelo agente A	166
7.6	Segunda EnO da Aliança.	167
7.7	Proposta de reorganização do agente E_1 com foco na ED.	171
7.8	Proposta de reorganização do agente E_2 com foco na EO.	171
7.9	Proposta de reorganização do agente E_3 com foco na EO.	172
7.10	Proposta de reorganização do agente B com foco na EF.	173
8.1	Tela do simulador TeamBots.	177
8.2	Especificação organizacional do JOJTEAM.	179
8.3	Exemplo de área dos jogadores.	180
8.4	Arquitetura geral para o agente jogador do JOJTEAM.	182
8.5	Camada deliberativa do agente jogador.	183
8.6	Exemplo de algoritmo para elaboração de plano de reorganização.	185
8.7	Resultados do aprendizado do time com estado do mundo incluindo tempo e placar.	191
8.8	Resultados do aprendizado do time com estado do mundo incluindo somente o tempo.	193
A.1	Exemplo de interação no SACI (HÜBNER; SICHMAN, 2000).	203
A.2	Exemplo de anúncio de habilidades no SACI (HÜBNER; SICHMAN, 2000).	204

Lista de Tabelas

2.1	Comparação entre algumas propostas de modelos organizacionais	35
3.1	Comparação entre os modelos MOISE e o $MOISE^+$	80
5.1	Classificação de alguns trabalhos sobre reorganização de SMA	115
6.1	Relações deônticas para o processo de reorganização	128
6.2	Papéis e missões assumidos pelos agentes do exemplo escola	140
6.3	Resumo da história da entidade organizacional escola	140
7.1	Exemplo de adoção de papéis, missões e metas na Aliança	169
7.2	Resumo da história da entidade organizacional Aliança	173
7.3	Papéis e missões dos agentes da Aliança	174
7.4	Classificação das propostas	174
7.5	Resultado das eleições	175
8.1	Política aprendida pelo JOJTEAM utilizando estado do mundo formado por tempo e placar	192

Lista de Abreviaturas

AOSE Agent Oriented Software Engineering

B2B Business to Business

ED Especificação Deontica

EE Especificação Estrutural

EF Especificação Funcional

EnO Entidade Organizacional

EO Especificação Organizacional

ES Esquema Social

IA Inteligência Artificial

IAD Inteligência Artificial Distribuída

KQML Knowledge Query and Manipulation Language

LCA Linguagem de Comunicação entre Agentes

MDP Processo de Decisão de Markov

MOISE Model of Organization for multi-agent SystEms

OO Orientação a Objetos

RDP Resolução Distribuída de Problemas

SMA Sistemas Multiagentes

TAEMS Task Analysis, Environment Modeling, and Simulation

UML Unified Modeling Language

Convenções e Lista de Símbolos

Na notação das fórmulas, as seguintes convenções são utilizadas:

- letras maiúsculas são conjuntos, exemplo: $\mathcal{P}, \mathcal{SG}$;
- letras minúsculas são elementos de conjunto, exemplo: $\rho \in \mathcal{R}, gt \in \mathcal{SG}$;
- subscrito diferencia um elemento de um conjunto de outro, exemplo: $\rho_{aluno}, \rho_{professor}$;
- $=_{\text{def}}$ define um conjunto;
- $\mathbb{P}(A)$ é o conjunto potência de A (o conjunto de todos os subconjuntos de A);
- $A \times B$ é o produto cartesiano dos conjuntos A e B ;
- A^* é o produto cartesiano de infinitos conjuntos A , ou seja $A^* = A \times A \times A \dots$;
- subscrito numérico a_n indica o n -ésimo elemento de uma tupla;
- $A \rightarrow B$ é uma função total tendo como domínio o conjunto A e imagem B ;
- $A \mapsto B$ é uma função parcial tendo como domínio o conjunto A e imagem B ;
- $a \mapsto b$ é um mapeamento da função $A \rightarrow B$, tal que $a \in A$ e $b \in B$;
- $F \oplus F'$ é a sobreposição da função F pela função F' . Sendo F e F' funções do mesmo tipo, o operador \oplus gera uma nova função a partir da união dos mapeamentos de F e F' . Para os elementos do domínio onde ambas possuem mapeamento, prevalece o mapeamento de F' . Por exemplo

$$\{a \mapsto b, c \mapsto d\} \oplus \{a \mapsto x, f \mapsto g\} = \{a \mapsto x, c \mapsto d, f \mapsto g\}$$

- $\#S$ cardinalidade de S (o número de elementos do conjunto S);
- \Rightarrow , implicação lógica;
- \wedge , conjunção lógica;
- \vee , disjunção lógica;
- \neg , negação lógica;
- \forall , quantificador universal, $\forall x \bullet \theta$ lê-se “para todo x tal que θ ”;
- \exists , quantificador existencial, $\exists x \bullet \theta$ lê-se “existe x tal que θ ”.

Os seguintes símbolos são definidos:

símbolo	descrição	página
\sqsubset	relação de herança entre papéis	41
\bowtie	relação de compatibilidade entre papéis	43
\prec	relação de preferência entre duas missões	64
γ	objetivo da organização	71
A	conjunto de ações no ambiente	37
\mathcal{A}	conjunto de agentes	71
B	conjunto de comportamentos possíveis	37
\mathcal{C}	conjunto de compatibilidade entre papéis	43
\mathcal{DS}	conjunto de todas as Especificações Deônticas	68
E	conjunto dos comportamentos do SMA permitidos pelo ambiente onde está inserido o SMA	38
F	conjunto dos comportamentos do SMA permitidos pelos esquemas da organização do SMA	38
\mathcal{FS}	conjunto de todas as Especificações Funcionais	65
\mathcal{G}	conjunto de metas (objetivos)	57
\mathcal{GI}	conjunto de instâncias de grupo	71
\mathcal{GT}	conjunto de todas as especificações de grupos de uma especificação estrutural	45
\mathcal{GV}	conjunto de valores para as metas	56
\mathcal{L}	conjunto de ligações entre papéis	42

continua. . .

símbolo	descrição	página
\mathcal{M}	conjunto de missões	61
\mathcal{O}	conjunto de observações do ambiente	37
\mathcal{OS}	conjunto de todas as Especificações Organizacionais	38
\mathcal{P}	conjunto dos comportamentos do SMA que correspondem à finalidade do SMA	38
\mathcal{P}	conjunto de planos	61
\mathcal{PR}	conjunto de preferência de missões	64
\mathcal{R}	conjunto de papéis, os elementos do conjunto são denotados por ρ	40
\mathcal{RG}	conjunto das especificações de grupo raiz de uma EE	48
\mathcal{S}	conjunto dos comportamentos do SMA permitidos pela estrutura da organização do SMA	38
\mathcal{SCH}	conjunto de esquemas sociais	65
\mathcal{SI}	conjunto de instâncias de esquemas sociais	71
\mathcal{SS}	conjunto de todas as Especificações Estruturais	48

1 Introdução

Tanto a Ciência da Computação quanto a Inteligência Artificial (IA) têm buscado formas de conceber sistemas que se aproximam da realidade considerando, em geral, as visões que outras áreas do conhecimento têm desta mesma realidade. Assim surgiram a orientação a objetos (da Matemática), a representação de conhecimento e raciocínio (da Psicologia e da Lógica), as redes neurais (da Biologia), etc. De forma análoga, a área de Sistemas Multiagentes (SMA) é influenciada pela Sociologia e, portanto, tem vislumbrado uma concepção de sistemas com propriedades que até então somente sociedades possuíam. Trouxe também novos problemas e desafios, um dos quais será assunto desta tese.

1.1 Sistemas Multiagentes

A motivação inicial para esta tese provém da principal característica dos SMA que, ao contrário dos paradigmas tradicionais da IA, têm como objeto de estudo a *coletividade* e não um único indivíduo. Desta forma, deixam de ter atenção as iniciativas, seja mental (IA simbolista) ou neural (IA conexionista), de compreender e simular o comportamento humano isoladamente, passando o foco da atenção para a forma de interação entre as entidades que formam o sistema (chamadas de agentes) e para a sua organização. Este paradigma é motivado pela observação de alguns sistemas naturais, nos quais se percebe o surgimento de um comportamento inteligente a partir da interação de seus elementos (JOHNSON, 2001). Por exemplo, apesar de uma colônia de formigas ser formada por seres simples, pode-se dizer que o formigueiro como um todo é um sistema complexo cujo comportamento é mais inteligente do que os das formigas que o formam; os neurônios são células simples, mas de sua interação e organização emerge um comportamento complexo e inteligente. Estes dois exemplos mostram que a co-

letividade possui características que não podem ser reduzidas aos componentes que a formam, mas que são essenciais para o comportamento bem adaptado que tais sistemas apresentam.¹

A área de SMA estuda o comportamento de um grupo *organizado* de agentes *autônomos* que cooperam na resolução de problemas que estão além das capacidades de resolução de cada um individualmente. Duas propriedades, aparentemente contraditórias, são fundamentais para os SMA: a autonomia dos agentes e sua organização (BRIOT; DEMAZEAU, 2002). O atributo autônomo significa aqui o fato de que um agente tem sua existência independente dos demais e mesmo do problema sendo solucionado (WEISS, 1999, p. 548)². Por outro lado, a organização estabelece restrições aos comportamentos dos agentes, visando estabelecer um comportamento grupal coeso. Muitas das propriedades desejadas nos SMA advêm do equilíbrio destes dois opostos. Portanto, compreender como estas duas propriedades interagem é uma questão importante (e interessante) no contexto dos SMA.

Tomando um ponto de vista de desenvolvimento de sistemas computacionais, o objetivo da área de SMA passa a ser a definição de modelos genéricos de agentes, interações e organizações que possam ser instanciados dinamicamente dado um problema (estas etapas são brevemente descritas na figura 1.1). Dado este ideal de metodologia de desenvolvimento de sistemas, esta abordagem apresenta as seguintes características (ALVARES; SICHMAN, 1997):

- os agentes são concebidos independentemente de um problema particular;
- a interação entre os agentes não é projetada anteriormente, busca-se definir protocolos que possam ser utilizados em situações genéricas;
- a decomposição de tarefas para solucionar um dado problema pode ser feita pelos próprios agentes;

¹Uma definição mais detalhada de SMA, seus problemas e aplicações podem ser encontradas nas seguintes referências (BORDINI; VIEIRA; MOREIRA, 2001; WOOLDRIDGE, 2002; FERBER, 1999; ALVARES; SICHMAN, 1997; WEISS, 1999; DEMAZEAU; MÜLLER, 1990; JENNINGS; WOOLDRIDGE, 1998).

²No caso, trata-se de uma autonomia de existência. Para funcionar, um agente não precisa de outros agentes, mesmo que para alcançar seus objetivos ele eventualmente precisará da ajuda de outros. Existem outras formas de autonomia, Castelfranchi (1990), por exemplo, define um agente autônomo como aquele que decide quais objetivos adotar. Outras definições podem ser encontradas em (SICHMAN, 1995).

- não existe um controle centralizado da resolução do problema.

Destas características decorrem algumas vantagens:

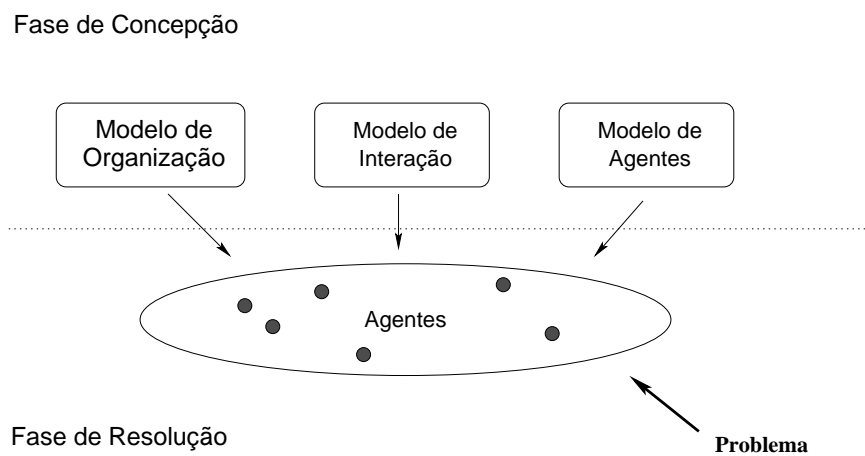
- Viabilizam sistemas *adaptativos* e *evolutivos*: o SMA tem capacidade de adaptação a novas situações, tanto pela eliminação e/ou inclusão de novos agentes ao sistema quanto pela mudança da sua organização.
- É uma *metáfora natural* para a modelagem de sistemas complexos e distribuídos: em muitas situações o conhecimento está distribuído, o controle é distribuído, os recursos estão distribuídos. Quanto à modelagem do sistema, a decomposição de um problema e a atribuição dos sub-problemas a agentes permite um alto nível de abstração e independência entre as partes do sistema (JENNINGS; WOOLDRIDGE, 1998).
- Tira proveito de ambientes *heterogêneos* e *distribuídos*: agentes com arquiteturas diferentes, que funcionam em plataformas diferentes, distribuídas em uma rede de computadores, podem cooperar na resolução de problemas. Isto permite o uso das potencialidades particulares de cada arquitetura e, pela distribuição, melhora o desempenho do sistema.
- Permite conceber *sistemas abertos*: os agentes podem migrar entre sociedades, isto é, agentes podem sair e entrar em sociedades, mesmo que desenvolvidos por projetistas e objetivos distintos. Tal abertura permite a evolução e a adaptabilidade do sistema³.

1.1.1 Agentes

Um componente indispensável em um SMA são os agentes. Dentre as definições possíveis para o termo agente (Franklin e Graesser (1997) discutem várias delas), a mais próxima do cenário apresentado acima é a seguinte:

Um agente é uma entidade lógica ou física à qual é atribuída uma certa missão que ela é capaz de cumprir de maneira autônoma e em coordenação com outros agentes. (BRIOT; DEMAZEAU, 2002)

³Esta noção de sistema aberto (baseada, entre outros, em (COSTA; HÜBNER; BORDINI, 1994; HÜBNER, 1995; BORDINI, 1994; BORDINI, 1999)) difere da noção adotada, por exemplo, na área de engenharia de software, onde a principal característica de um sistema aberto é seguir determinados padrões.



De forma geral, o ciclo de vida de um SMA passa por duas etapas: concepção e resolução (SICHTMAN, 1995). Na concepção são definidos modelos de propósito geral para os agentes, para suas interações e para suas formas de organização. Na resolução, um grupo de agentes adota estes modelos para resolver os problemas que lhe são apresentados. Diferentes tipos de problemas demandam dos agentes diferentes escolhas de modelos. A principal característica é a independência entre a concepção dos modelos e o problema, isto é, os modelos não são desenvolvidos para solucionar um problema particular. Por exemplo, os protocolos contratuais de Smith (1980) são um modelo de interação aplicável em vários tipos de problemas.

Figura 1.1: Desenvolvimento ideal de Sistemas Multiagentes.

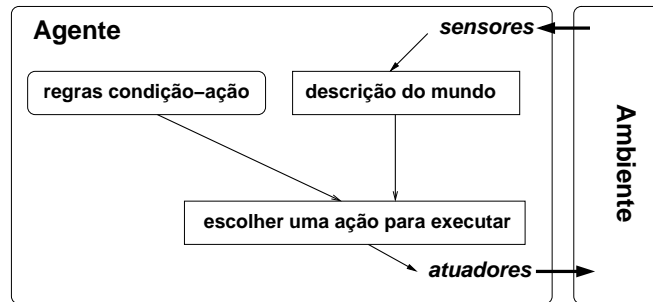
O comportamento apresentado pelos agentes para cumprir suas missões normalmente é explicado seguindo dois grandes modelos de funcionamento interno: agentes *reativos* e agentes *cognitivos*⁴. Os agentes reativos têm um comportamento muito simples: escolhem suas ações baseados unicamente nas percepções que têm do ambiente (na figura 1.2(a) é descrita uma possível arquitetura⁵, entre outras, para este modelo de funcionamento). Considerando este funcionamento bastante simples, este tipo de agente apresenta em geral as seguintes propriedades: normalmente possui representação de conhecimento implícita no código; por não possuir memória, não tem história dos fatos que aconteceram e das ações que executou; não tem controle deliberativo (planejado) de suas ações; em geral, formam organizações do tipo etológico; e as sociedades são formadas por muitos agentes. Nestas sociedades, o interesse está voltado para a emergência de um comportamento global a partir da interação de um grupo grande de agentes (como no exemplo da colônia de formigas).

Nos modelos cognitivos, normalmente os agentes possuem um *estado mental*⁶ e funcionam racionalmente, isto é, raciocinam para construir um plano de ações que leva a um objetivo pretendido (figura 1.2(b)). Portanto, apresentam características particulares que os diferenciam de programas convencionais e dos agentes reativos. Dentre elas, convém enumerar as mais citadas nos trabalhos de SMA: têm autonomia funcional pois podem alterar seu funcionamento a fim de adaptar-se melhor ao seu ambiente (RAO; GEORGEFF, 1995); estão continuamente em funcionamento; são sociáveis (possuem a capacidade de comunicação e de modelagem dos outros); possuem representação de conhecimento explícita no código (conhecimento introspectivo); o mecanismo de controle é deliberativo (o agente raciocina sobre que ações realizar); têm memória e portanto podem lembrar de ações realizadas no passado; e normalmente, as sociedades são formadas

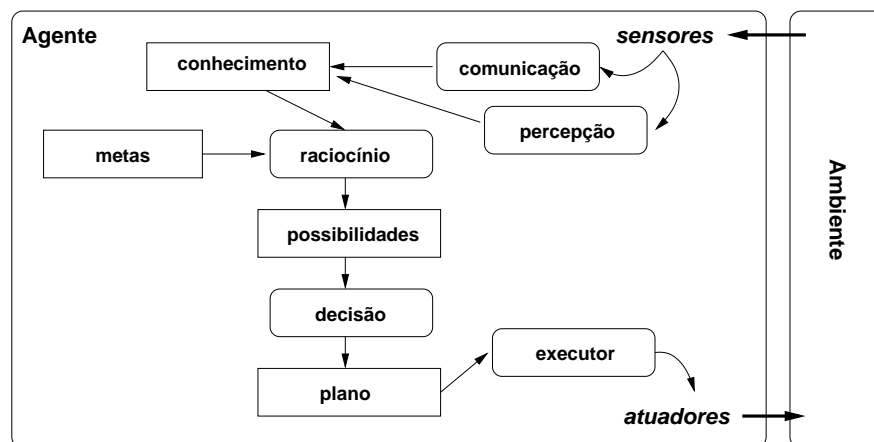
⁴Boissier (2002) propõe uma taxonomia mais detalhada para os modelos de agentes. São propostos três modelos gerais: agentes autônomos, interativos (com capacidade de interação com os demais agentes) e sociais (com capacidade de raciocínio sobre os demais agentes). Em cada um destes três modelos, há propostas de (sub) modelos reativos e cognitivos.

⁵Enquanto um modelo *explica* o comportamento de um sistema, a arquitetura diz como esse sistema opera. Evidentemente, um sistema pode ser explicado segundo vários modelos e um mesmo modelo pode ter várias arquiteturas que, por sua vez, podem ter várias implementações computacionais.

⁶Estados mentais são representações mentais que estão dirigidas para objetos ou “coisas” no mundo (COHEN; LEVESQUE, 1987; SEARLE, 1983). Crer que está chovendo é uma representação mental de um certo tipo (crença) sobre algo do mundo (estar chovendo). Outros tipos de estados mentais são, por exemplo, desejo, intenção, temor, dúvida e ódio.



(a) Agente Reativo



(b) Agente Cognitivo

Em (a) a arquitetura proposta por Russel e Norvig (1995) para o modelo reativo de agente é ilustrada. O comportamento do agente é determinado unicamente pela percepção do ambiente e por um conjunto fixo de regras. Normalmente se pode dizer de um agente reativo: “o agente percebeu y e portanto fez x ”. Para o modelo cognitivo, tem-se em (b) a arquitetura proposta por Demazeau e Müller (1990): o conhecimento que o agente possui é formado a partir da sua percepção do ambiente e da comunicação com outros agentes. Dado este conhecimento e uma meta (outro tipo de estado mental), o agente gera um conjunto de possíveis planos que atingem esta meta. Dadas estas possibilidades, o agente delibera sobre o melhor plano a ser executado. Aqui se pode dizer que “o agente fez x porque tem por objetivo y e x faz parte de um plano que leva à satisfação de y ”. Obviamente esta diferença entre as arquiteturas somente pode ser percebida observando-se a arquitetura interna dos agentes, pois somente pela observação do comportamento não se pode dizer se um agente é reativo ou cognitivo.

Figura 1.2: Duas arquiteturas clássicas de agentes.

por poucos agentes. Um agente não precisa necessariamente apresentar todas estas características para ser considerado cognitivo pois, como já foi comentado, não há consenso sobre a definição de agente.

1.1.2 Organização dos agentes

Um grupo de agentes pode adotar (ou ser submetido a) diferentes formas de organização e certamente esta opção terá conseqüências no desempenho do grupo (GASSER, 2001). Não há, porém, como se determinar previamente se uma organização terá um rendimento ótimo em qualquer circunstância (GALBRAITH, 1977). Vários fatores determinam o desempenho da organização de um sistema: o tipo das tarefas sendo realizadas pelos agentes, suas capacidades cognitivas, seu treinamento, a constância do ambiente, restrições legais ou políticas e o tipo de resultado esperado (eficiência, eficácia, custo mínimo, precisão, etc.). A área de estudo que se dedica a este problema, denominada teoria organizacional, é definida por Prietula, Carley e Gasser (1998) da seguinte forma:

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

Em um SMA, esta teoria é utilizada pelo projetista do sistema ou pelos próprios agentes para determinar qual a melhor organização a ser adotada (como sugerido na figura 1.1). Particularmente, é interessante que a sociedade consiga mudar sua organização durante seu funcionamento a fim de adequar-se melhor às mudanças em seu ambiente. Para um SMA ter a capacidade de (auto) organizar-se, Pattison, Corkill e Lesser (1987), So e Durfee (1993) sugerem que ele deva implementar as seguintes tarefas:

Monitoração: esta tarefa calcula a eficiência da sociedade considerando um conjunto de parâmetros e determina quando uma reorganização é necessária.

Projeto de organizações: consiste em especificar novas formas de organização adequadas à nova situação.

Avaliação e seleção: consiste em escolher uma das organizações especificadas. Envolve a avaliação de cada alternativa gerada na etapa anterior conside-

rando uma medida de desempenho, isto é, qual a ganho esperado se tal organização for escolhida.

Implementação: esta etapa estabelece a nova organização da sociedade atual.

Deve-se considerar o fato de que a sociedade está funcionando e a troca de organização tem que se dar à medida que as estruturas da organização anterior não são mais necessárias.

Para cada uma das tarefas acima há várias perguntas sem respostas ou com múltiplas respostas não consensuais, tais como:

- i)* O que é uma organização exatamente? O que a constitui? Que aspectos são relevantes e devem ser modelados? Como descrevê-la? Onde esta descrição é armazenada (na memória dos agentes, em uma memória social)?
- ii)* Que propriedades um SMA deve possuir para facilitar seu processo de reorganização?
- iii)* Como identificar a necessidade de alterações organizacionais? Como a avaliação da organização é feita? Quais os parâmetros? Como estes dados são obtidos?
- iv)* Quem deve realizar a reorganização, um único agente com autoridade para isso, todos os agentes, ou um grupo especializado nesta tarefa?
- v)* Como projetar organizações para uma determinada finalidade?
- vi)* Como estimar o ganho de adotar uma organização em um dado ambiente?
- vii)* Como e por que agentes autônomos irão se submeter a uma (re)organização? Que restrições devem existir na arquitetura dos agentes para que eles possam perceber e se comportar adequadamente a uma mudança organizacional? Se o fenômeno da emergência impõe restrições/propriedades no nível social a partir do comportamento individual dos agentes, qual o fenômeno dual, ou seja, do nível social para o individual (CONTE; GILBERT; SICHMAN, 1998)?

1.2 Objetivos

Responder a todas as perguntas enumeradas acima certamente está fora do alcance deste trabalho, que irá ater-se às questões de reorganização. O *objetivo principal* desta tese é apresentar um modelo de reorganização para SMA abertos que considere mudanças em seus aspectos organizacionais mais relevantes e que possa ser realizado pelos próprios agentes. Contudo, o processo de reorganização somente pode ser compreendido a partir de uma definição clara de como uma organização pode ser descrita, para que tal descrição possa ser alterada quando necessário. Adicionalmente, o modelo segundo o qual uma organização é descrita irá facilitar ou dificultar sua reorganização.

Portanto, os seguintes objetivos específicos são colocados:

1. a elaboração de um modelo de organização identificando os elementos mais relevantes de uma organização;
2. a criação de uma linguagem formal para especificação de organizações de forma que tanto os agentes quanto os projetistas do SMA possam utilizá-la;
3. a especificação de um processo de reorganização sobre o modelo organizacional proposto; e
4. a identificação das propriedades que os agentes e/ou sua plataforma de funcionamento devem possuir para viabilizar o processo de reorganização.

1.3 Organização do texto

O texto desta tese está organizado em três partes (conforme o ilustra o diagrama da figura 1.3):

Organização de SMA. Antes de compreender o que é reorganização, é necessário compreender o que é organização, particularmente, organização de um SMA. A visão que outros autores têm da organização dos SMA é apresentada no capítulo 2. A partir da análise destas visões, pôde-se escolher um modelo de base, denominado MOISE (HANNOUN, 2002), e estendê-lo adicionando propriedades importantes para o processo de reorganização. O

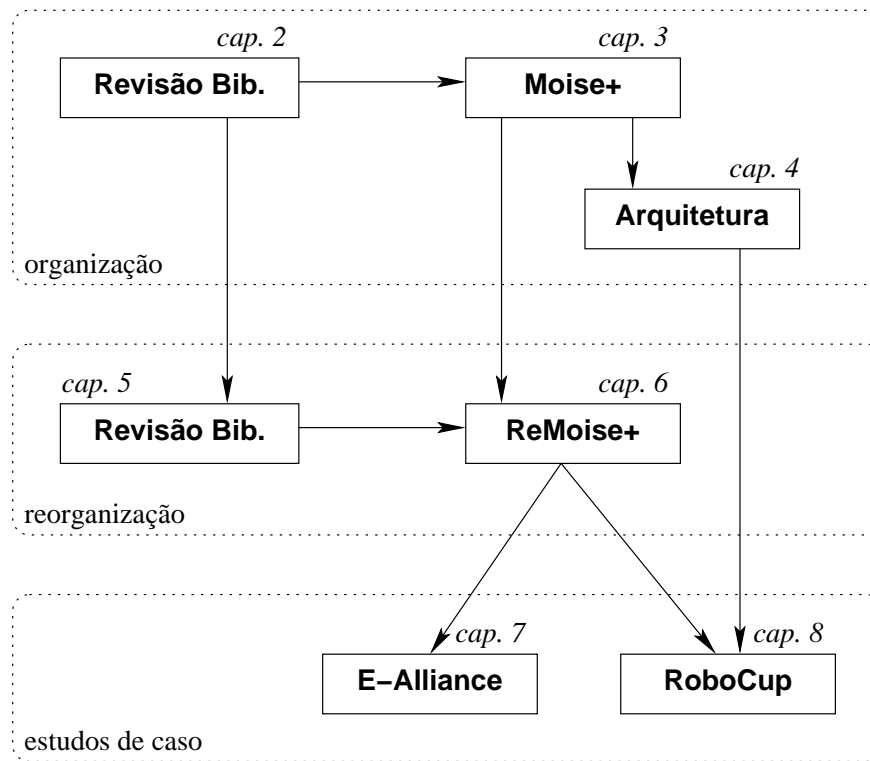


Figura 1.3: Dependência entre os capítulos.

resultado desta extensão é o modelo organizacional \mathcal{MOISE}^+ , apresentado no capítulo 3. O capítulo 4 detalha o modelo, propondo uma arquitetura de funcionamento para um SMA organizado conforme o \mathcal{MOISE}^+ .

Reorganização de SMA. De forma idêntica à parte anterior, no capítulo 5 o problema de reorganização é bem caracterizado e, em seguida, são sintetizadas as visões e as propostas de outros autores para tal problema. Com base nestas propostas, no capítulo 6 se descreve como um SMA pode se reorganizar utilizando um processo de reorganização definido por meio do modelo \mathcal{MOISE}^+ .

Estudos de caso. Foram desenvolvidos dois estudos de casos, nos domínios de empresas virtuais (capítulo 7) e de futebol de robôs (capítulo 8). Convém ressaltar que estes estudos de caso, além de demonstrar a viabilidade de implementação da proposta de reorganização, têm por objetivo a sua avaliação. Não se pretende desenvolver soluções finais para os problemas dos estudos de caso. Por exemplo, não é objetivo desta tese desenvolver soluções para empresas virtuais.

Por fim, o capítulo 9 contém a síntese dos resultados alcançados frente aos objetivos propostos e a descrição de alguns trabalhos futuros que se tornaram possíveis a partir da pesquisa desenvolvida.

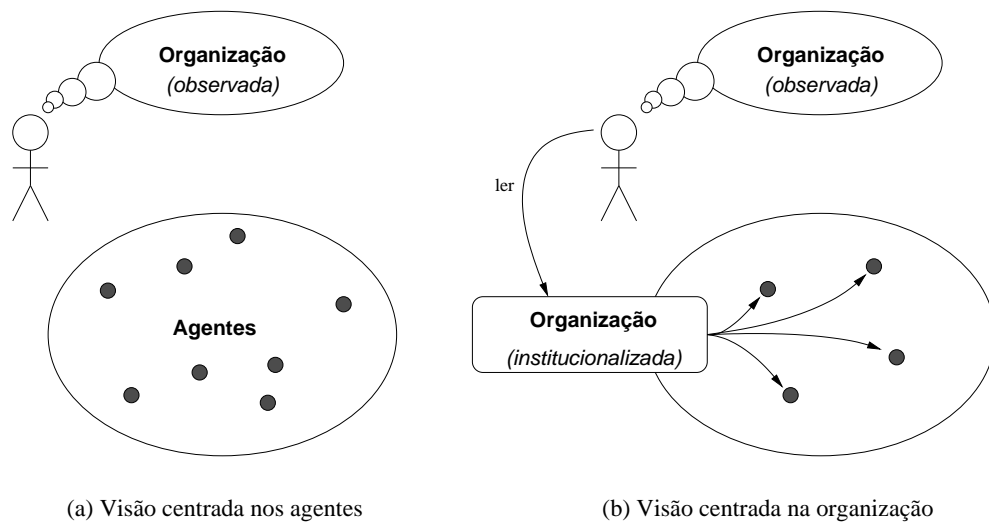
Parte I

Organização

2 Organização em Sistemas Multiagentes

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, a organização de um grupo de onze pessoas jogando futebol é que o diferencia de um grupo de onze pessoas quaisquer, uma empresa é um sistema que somente existe como sendo organizado, até mesmo o conceito de “ser vivo” passa pelo conceito de organização (ROMESÍN; GARCÍA, 1997). A partir destes exemplos, é possível inferir o propósito principal que a organização tem nos sistemas: fazer com que sua finalidade seja facilmente mantida.¹ Enquanto parece simples caracterizar um sistema como mais bem organizado com base no modo como mantém e busca sua finalidade, definir o que é uma organização, o que a constitui, quais suas formas e estruturas não é uma tarefa tão simples. Porém, a definição precisa da constituição de uma organização é pré-condição para que o projetista do SMA ou o grupo de agentes que lhe dá forma possam raciocinar sobre ela, sobre seu desempenho e, eventualmente, tomar a iniciativa de alterá-la. Antes de construir tal definição, convém avaliar o que outros autores já têm proposto neste assunto. Assim, este capítulo propõe uma classificação de tipos de organização e como os trabalhos de alguns autores enquadram-se dentro desta visão. Busca-se deste modo tornar mais claro como os elementos que formam uma organização estão relacionados com a sua finalidade.

¹O termo finalidade será utilizado em um sentido mais amplo que o termo objetivo comum. Enquanto um objetivo normalmente se refere a um estado do ambiente que é desejado pelo sistema e, quando alcançado, o sistema pode parar, a finalidade tem uma noção de continuidade, um funcionamento que se deseja manter e não um estado que se deseja chegar. Por exemplo, uma célula tem por finalidade se manter viva e para isso vários “objetivos” são constantemente buscados.



Em (a) a organização existe (explícita ou implicitamente) somente dentro dos agentes, um observador do SMA não tem acesso a ela e somente pode criar uma visão subjetiva de tal organização a partir da observação do comportamento do sistema. Em (b), além da organização observada, existe uma descrição explícita da organização e tal descrição encontra-se representada externamente. Um observador do sistema pode, portanto, conhecer a organização do sistema tanto consultando tal descrição quanto contruindo uma descrição subjetiva por meio de observação.

Figura 2.1: Pontos de vista para organização de um SMA.

2.1 Introdução

A literatura de SMA apresenta várias definições para organização. Algumas destas são apresentadas a seguir, classificadas segundo a sugestão de Lemaître e Excelente (1998). Estes autores propõem a existência de dois pontos de vista para a compreensão do que é a organização de um grupo de agentes: visão *centrada nos agentes* e *centrada na organização*. Na primeira abordagem, o SMA não possui uma representação explícita de sua organização. Um observador ou um agente da sociedade somente pode inferir uma descrição subjetiva da organização, isto é, uma descrição *construída* por ele mesmo a partir da observação do comportamento dos agentes de tal sociedade (figura 2.1). Esta descrição subjetiva da organização será chamada de *organização observada*. Por exemplo, pode-se apenas *observar* a organização de um formigueiro, não se pode obter uma descrição única da sua real organização porque tal organização está distribuída e implícita no DNA das várias formigas do formigueiro.

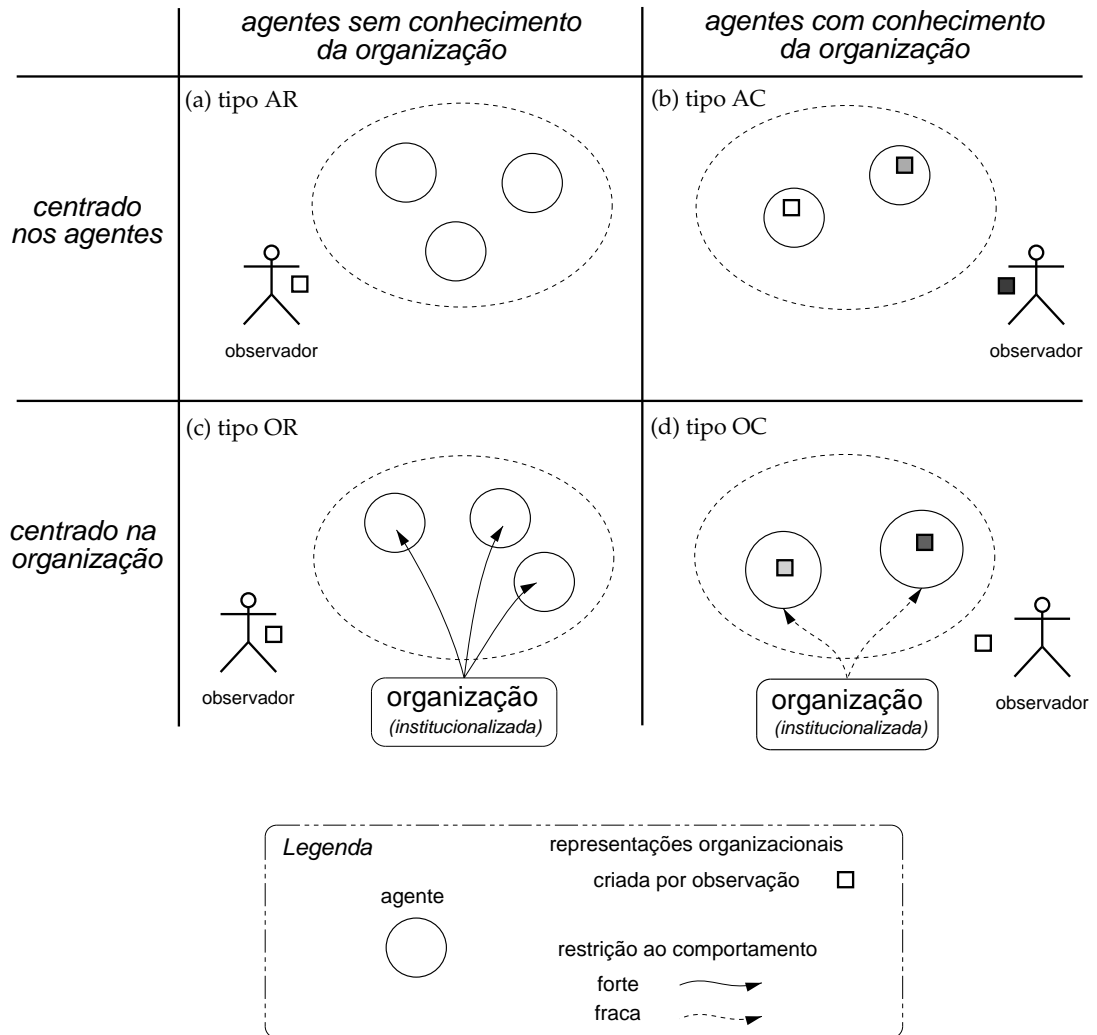
No segundo ponto de vista, a organização existe objetivamente, isto é, o observador pode *obter* uma descrição da organização que a sociedade está adotando sem precisar observar seu comportamento ou mesmo considerar os agentes que a compõem. Esta descrição será chamada de *organização institucionalizada*. Por exemplo, em uma escola a descrição de sua organização existe fora dos agentes na forma de manuais de procedimentos, organogramas, etc. Mesmo possuindo uma organização institucionalizada, um observador deste segundo tipo de sociedade (eventualmente um dos seus agentes) pode construir uma descrição subjetiva desta última a partir da observação de seu comportamento.²

Se do ponto de vista de um observador externo ao SMA podem existir duas situações, do lado dos agentes que formam a sociedade também podem ser concebidas duas situações: uma onde os agentes são capazes de explicitamente representar uma organização e outra onde os agentes não são capazes de fazê-lo. O fato dos agentes terem ou não uma representação interna de sua organização não implica que sua sociedade tenha ou não uma organização institucionalizada. Portanto, propõe-se aqui estender a classificação proposta por Lemaître e Excellence (1998) para quatro tipos de organização: AR, AC, OR e OC. Estes tipos estão ilustrados na figura 2.2 e descritos nas próximas seções.

2.1.1 Organização do tipo AR

Os SMA com organização do tipo AR são concebidos por uma visão centrada nos agentes e tais agentes não representam e, conseqüentemente, não raciocinam sobre a organização observada. Os sistemas formados por agentes reativos são um primeiro tipo de sistema onde se pode perceber claramente a visão na qual a organização não existe explicitamente no sistema mas existe somente como uma construção feita a partir do seu comportamento. Este tipo de sistema é formado por agentes muitos simples, em geral com comportamentos básicos disparados pela percepção que têm do ambiente. O desenvolvimento do sistema consiste na definição de comportamentos elementares e espera-se que a partir deles, por meio da interação entre os agentes e o ambiente, venham a surgir (emergir) comporta-

²Esta possibilidade contribui para a complexidade da representação da organização pois o processo de construção da organização observada é subjetivo e portanto não é único: diferentes agentes podem construir descrições organizacionais distintas a partir da observação do mesmo sistema. Por esta razão, vários sistemas criam uma descrição institucionalizada (objetiva), procurando minimizar estas diferenças.



Em (a) não existe organização dentro do SMA, apenas um observador pode modelar o comportamento do sistema como sendo organizado. Em (b), a organização que existe no SMA é construída pelos próprios agentes a partir da observação do sistema. Por serem construções subjetivas, cada um dos agentes pode construir uma representação organizacional distinta. Na figura, estas diferenças são representadas por quadrados de tons diferentes. Em (c), o sistema possui uma organização, mas os agentes não têm capacidade para conhecer tal organização. E, por último, em (d) tem-se que os agentes podem tanto conhecer a organização institucionalizada quanto a organização observada. Novamente, os agentes podem ter representações distintas devido ao seu processo de percepção.

Figura 2.2: Tipos de organização.

mentos complexos. Particularmente, deseja-se que o sistema venha a apresentar uma organização (STEELS, 1990). Por exemplo, no sistema MANTA (DROGOUL; CORBARA; LALANDE, 1995), onde é simulada a gênese de uma colônia de formigas, observou-se que o formigueiro apresentou estratégias de controle populacional e mecanismos de divisão do trabalho, ainda que tais comportamentos não tenham sido intencionalmente programados no código das formigas. Um observador do formigueiro pode perfeitamente dizer que as formigas estão organizadas, mesmo que esta organização não tenha sido deliberadamente construída pelas formigas. Evidentemente, esse tipo de organização emergente impossibilita os agentes da sociedade alterarem intencionalmente sua organização, já que não a conhecem.³

2.1.2 Organização do tipo AC

Os SMA com organização do tipo AC também são concebidos por uma visão centrada nos agentes, contudo tais agentes têm capacidade de representar e raciocinar sobre a organização observada. Neste caso, os agentes raciocinam sobre uma representação interna de sua organização construída a partir de percepção ou comunicação. Por exemplo, no trabalho de Cohen e Levesque (1991) sobre *Joint Comittment*, os agentes têm um estado mental de compromisso conjunto que restringe os comportamento dos agentes e é construído por meio de comunicação com os demais agentes.

2.1.3 Organização do tipo OR

Os SMA com organização do tipo OR são concebidos por uma visão centrada na organização e os agentes não têm capacidade de gerar uma representação de sua organização. Apesar da sociedade ter uma descrição de sua organização, os agentes não a representam internamente de modo explícito. Contudo, os agentes têm seu comportamento parcialmente determinado pela organização estabelecida. Por não terem uma representação da organização, eles não têm capacidade de raciocinar sobre ela e tirar proveito deste conhecimento na realização de suas tarefas. Quem conhece a organização e faz uso dela para determinar o comportamento

³Certamente existem sistemas onde ocorrem alterações não intencionais da organização por meio de mecanismos como, por exemplo, a *autopoiese* (ROMESÍN; GARCÍA, 1997). Este tipo de reorganização não intencional não será considerado neste trabalho.

dos agentes é, por exemplo, o projetista do sistema (BAEIJIS, 1998).

O tipo de restrições que esse tipo de organização impõe aos agentes será chamado de *restrição forte*, já que os agentes não têm como se comportar de forma diferente daquela estabelecida pela organização.⁴

2.1.4 Organização do tipo OC

Os SMA com organização do tipo OC também são concebidos por uma visão centrada na organização, porém os agentes têm capacidade de gerar uma representação interna de sua organização. Os agentes conhecem sua organização, têm seu comportamento parcialmente determinado por ela, sabem como a organização interfere no seu comportamento e no dos demais agentes e utilizam estas informações para melhorar seu funcionamento. Neste tipo e no tipo AC, os agentes podem, por exemplo, assumir o lugar de observador da figura 2.1 (a) e construir uma descrição subjetiva de sua organização. Entretanto, somente no tipo OC os agentes podem assumir o lugar do observador da figura 2.1 (b), podendo “ler” a organização institucionalizada. Neste último caso, porém, há que se fazer uma ressalva: enquanto observador do sistema, o agente pode construir uma organização observada distinta daquela obtida objetivamente (figura 2.2 (d)).

O tipo de restrições que esse tipo de organização impõe aos agentes será chamado de *restrição fraca*, já que os agentes podem não seguir o comportamento que a organização estabelece. Ao contrário das organizações tipo OR, onde as restrições comportamentais não estão sujeitas aos processos deliberativos dos agentes, no tipo OC, as restrições comportamentais descritas na organização institucionalizada são seguidas (ou não) pelo agente porque ele decide (ou não) fazê-lo.

⁴Considerando apenas o critério da organização, poderia-se dizer que este tipo de sistema não é um SMA como apresentado na figura 1.1, isto é, um sistema onde os agentes escolhem uma organização. Trataria-se, portanto, de um caso de Resolução Distribuída de Problemas (RDP), onde o projetista do sistema decompõe o problema em sub-problemas e cria os agentes para resolvê-los (ALVARES; SICHMAN, 1997). Porém, a diferenciação entre RPD e SMA não considera apenas o viés da organização que está sendo considerado nesta classificação.

2.1.5 Autonomia organizacional

Em analogia com a autonomia funcional de um agente (funcionar de modo diferente do que o projetista do agente lhe estabeleceu inicialmente), a *autonomia organizacional* de um agente em um SMA é definida como sua capacidade de se comportar de modo diferente do estabelecido pela organização. A visão organizacional apresentada permite situar este tipo de autonomia em três casos:

- para as organizações dos tipos AC e AR, não faz sentido identificar se há ou não autonomia organizacional pois não existem restrições organizacionais explicitamente definidas;
- para o tipo OR, os agentes não tem autonomia organizacional. A arquitetura do agente (e a implementação decorrente) não permitem que o agente se comporte de forma diferente da estabelecida pela organização; e
- no tipo OC, onde a organização institucionalizada estabelece uma restrição fraca no comportamento dos agentes, os agentes *podem* ter autonomia organizacional. Em organizações do tipo OC a autonomia organizacional é uma possibilidade, porém não uma necessidade. O fato de conhecer a organização é condição para um agente ter autonomia organizacional, contudo, o agente deve ter capacidade e razões para exercer tal autonomia.⁵

Dentre estas quatro formas de conceber a organização em um SMA, o tipo OC é o mais adequado no caso onde se deseja que os próprios agentes alterem sua organização, pois a organização corrente está claramente estabelecida e os agentes têm capacidade de raciocinar sobre ela.

Independentemente da forma com que os agentes ou os observadores percebem a organização do sistema, consideram-se as seguintes *premissas*:

⁵Convém aqui distinguir autonomia organizacional de reorganização. Na primeira, o agente pode se comportar de forma diferente da determinada pela organização. Na reorganização, o agente homologa uma nova forma de agir (uma nova organização) por meio do processo de reorganização. Em outras palavras, o processo de reorganização permite que o agente venha a agir de forma diferente sem incorrer em uma transgressão, já o uso da autonomia organizacional é uma transgressão.

- i*) todo SMA pode ser modelado como possuindo uma organização (seja objetiva ou subjetiva, institucionalizada ou observada); e
- ii*) todo agente tem parte do seu comportamento ditado pela organização (seja essa interferência conhecida ou não pelo agente).

A seguir são apresentados alguns trabalhos que lidam com os problemas organizacionais dos SMA. Como é possível perceber a noção de organização (AR, AC, OR ou OC) em qualquer trabalho na área de SMA (premissa (*i*)), o que torna praticamente todos os trabalhos da área candidatos a serem examinados, utilizou-se como critério de seleção o fato do trabalho ter a organização como objeto de estudo central ou por introduzir conceitos importantes na construção do modelo organizacional proposto nesta tese.

2.2 Definições centradas na organização

A visão centrada na organização apresenta algumas propriedades que a torna mais adequada em vários domínios de aplicação. Particularmente, ela é útil para o processo de reorganização (capítulo 5), razão pela qual este ponto de vista terá maior ênfase neste capítulo. Dentre estas propriedades, destacam-se:

- uma organização existe por mais tempo que seus agentes, e sua memória deve ser preservada independentemente dos agentes;
- não é uma tarefa simples obter uma descrição para a organização a partir dos seus agentes, principalmente em grupos de agentes reativos. Por exemplo, apesar das formigas estarem organizadas, não podemos saber precisamente que organização é esta, podemos apenas supor qual é. A falta de uma descrição da organização dificulta o raciocínio sobre tal organização e conseqüentemente sua reorganização;
- separando-se conceitualmente a organização da arquitetura interna dos agentes, torna-se possível raciocinar sobre a organização sem considerar a estrutura e o funcionamento dos agentes, isto é, pode-se projetar organizações em um nível de maior abstração. Por esta razão, esta visão é mais adequada em sistemas abertos, onde novos agentes podem entrar no sistema e não se conhece a arquitetura de tais agentes *a priori*.

A descrição explícita da organização de um SMA é feita segundo um modelo organizacional (uma “forma de ver” a organização), que determina, por exemplo, o que constitui a organização. Alguns modelos desta abordagem são descritos nas próximas seções.

2.2.1 Modelo organizacional AALAADIN

Um modelo de organização de SMA que representa bem o ponto de vista centrado na organização é o proposto por Ferber e Gutknecht (1998). Neste modelo, chamado de AALAADIN, 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 (figura 2.3). 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 comunicativa que assume papéis nos grupos onde é membro. Portanto, este modelo pode ser utilizado tanto para sociedades Tipo OR quanto Tipo OC. Essa propriedade é comum entre os modelos que seguem a visão centrada na organização.

Percebe-se que, segundo esta visão, surge a distinção entre tipos abstratos de agentes (representados pelos papéis) e instâncias destes tipos (os agentes reais e concretos), que correspondem a um aspecto *estático* e um aspecto *dinâmico* da organização. O termo utilizado por Ferber e Gutknecht para designar este aspecto estático de uma organização é estrutura organizacional. No AALAADIN, a estrutura organizacional é representada pelos grupos e papéis, o aspecto dinâmico surge no funcionamento da sociedade quando os agentes entram em grupos assumindo determinados papéis.

2.2.2 Modelo organizacional de Pattison, Corkill e Lesser

Um outro modelo de organização que também utiliza o conceito de estrutura organizacional é proposto por Pattison, Corkill e Lesser (1987), onde uma organização é vista como:

“um grupo de indivíduos que tem o propósito de realizar um conjunto de tarefas a fim de alcançar um conjunto de metas, observando

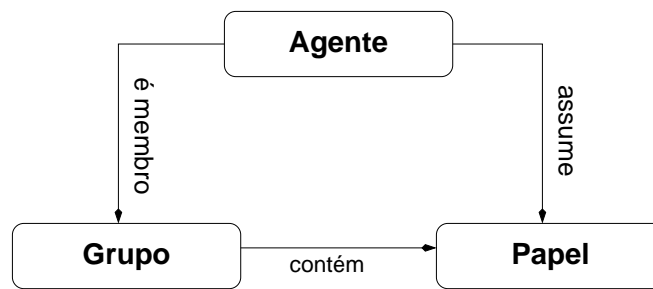


Figura 2.3: Modelo AALAADIN de organização segundo Ferber e Gutknecht (1998).

algumas restrições.” (PATTISON; CORKILL; LESSER, 1987)

A questão colocada pelos autores é a seguinte: dado um contexto e algumas descrições para possíveis estruturas organizacionais (aspecto estático), como determinar quais destas descrições podem ser instanciadas num determinado contexto (aspecto dinâmico). Para isso, uma linguagem de descrição de estruturas organizacionais é proposta. Esta linguagem considera os seguintes elementos na formação de uma organização: os componentes funcionais (agentes no modelo AALAADIN), as responsabilidades destes componentes (papéis no modelo AALAADIN), os recursos necessários para que os componentes funcionais atendam às suas responsabilidades (recursos de informação, ferramentas, consultores, etc.) e as relações entre os componentes (relações de autoridade e de comunicação). Na descrição de uma relação de comunicação, são estabelecidas as restrições para que a comunicação entre dois agentes aconteça, tais como: a linguagem de comunicação que deve ser utilizada, as informações que os agentes devem possuir, etc. A relação de autoridade determina com que importância um agente deve considerar uma mensagem recebida de outro agente. Conseqüentemente, estas relações definem uma hierarquia de autoridade no grupo.

2.2.3 Modelo organizacional TOVE

Fox et al. (1998) apresentam, através de um modelo organizacional chamado TOVE, uma definição mais detalhada para organizações no contexto empresarial (figura 2.4). As seguintes entidades são definidas:

Organização Uma organização 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 metas.

Papel Papéis são protótipos de funções a serem desempenhadas pelos agentes na organização. A cada papel são associadas as seguintes propriedades:

1. um conjunto de metas que o agente que assume o papel deve buscar,
2. um conjunto de processos que definem como as metas podem ser alcançadas (tais processos são descritos como uma estrutura de atividades),
3. um conjunto de autoridades que o agente necessitará para alcançar as metas (direitos sobre determinados recursos, por exemplo),
4. um conjunto de habilidades que o agente que pretende assumir o papel deve possuir,
5. um conjunto de restrições na execução dos processos, e
6. um conjunto de recursos necessários para o papel ser desempenhado.

Sobre os papéis também existem as seguintes relações:

1. hierarquia, um papel pode ser subordinado a outro (por exemplo quem assume o papel de chefe de divisão é subordinado a quem assume o papel de presidente da empresa), e
2. especialização, um papel pode especializar outro papel e herdar os direitos, obrigações, autoridades, etc. deste último (por exemplo, o papel de atacante especializa o papel de jogador).

Agente Um agente é membro de uma divisão da empresa, assume um ou mais papéis e pode se comunicar com outros agentes caso haja um ligação de comunicação entre eles. Os agentes também realizam atividades e podem consumir determinados recursos na realização destas atividades, porém estão sujeitos a determinadas restrições de comportamento.

Eventualmente os agentes podem formar times para realizar tarefas específicas. Um time tem um tempo de vida menor que uma divisão, sua existência termina quando a tarefa para o qual foi criado for concluída.

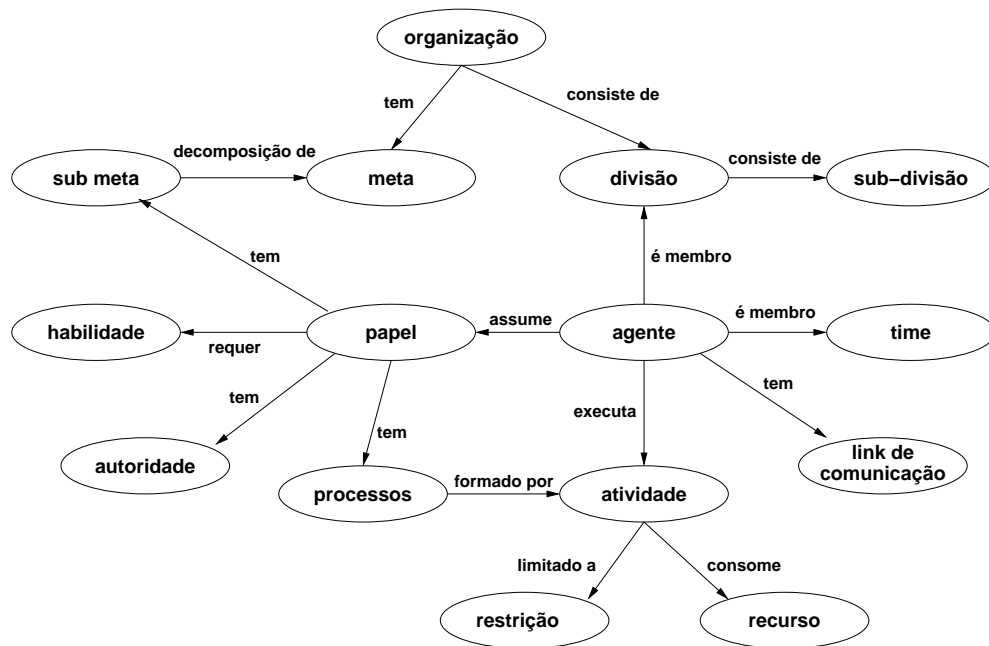


Figura 2.4: Taxonomia organizacional proposta por Fox et al. (1998).

2.2.4 Modelo organizacional TÆMS

Os modelos de organização vistos até aqui, particularmente o AALAADIN e o TOVE, procuram especificar vários dos atributos de uma organização, onde o conceito de papel social tem uma função central. Uma forma bastante distinta de conceber a organização é o modelo TÆMS (Task Analysis, Environment Modeling, and Simulation), no qual a noção central é a de *tarefa* (DECKER, 1998; DECKER; LESSER, 1994; DECKER, 1996; PRASAD et al., 1996). Como o nome sugere, o objetivo deste modelo é descrever a estrutura de tarefas de modo a viabilizar a análise e a simulação da organização.

As tarefas podem ser abordadas por três pontos de vista diferentes: objetivo, subjetivo e generativo. A visão *objetiva* considera a estrutura de tarefas completa, real, que resolve um problema em um determinado período de tempo. A visão *subjetiva* é tida pelos agentes que participam na execução da tarefa, ou seja, é a tarefa tal como vista pelos agentes. Em geral os agentes não vêem a tarefa como um todo, mas somente a parte que lhes cabe: aquela que a organização do sistema lhes permite. Apesar do modelo descrever a parte das tarefas visíveis aos agentes, a maneira como eles irão executar os métodos não é relevante. A visão *generativa* contém informações de como gerar várias visões objetivas e subjetivas para a

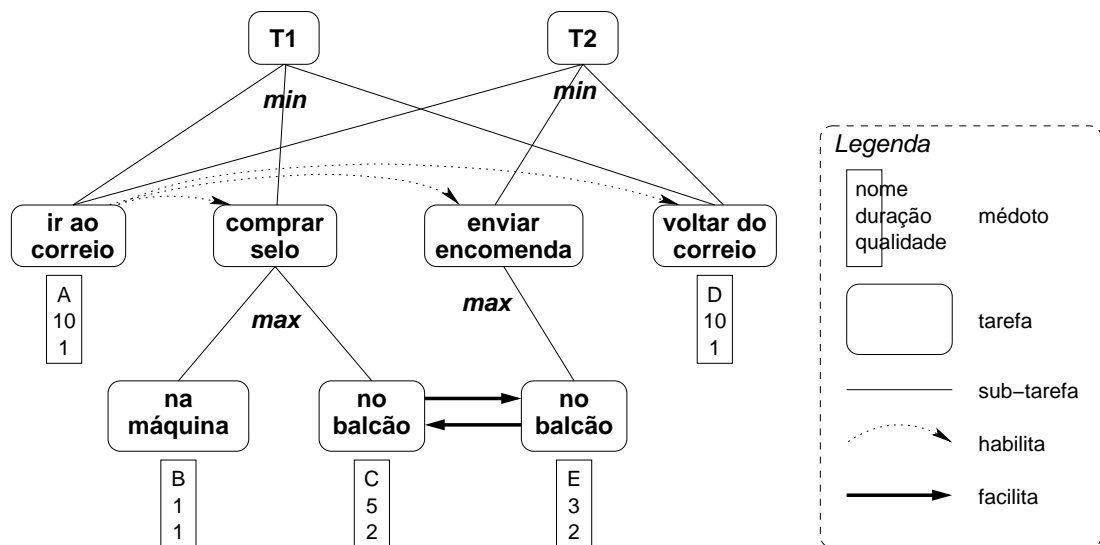
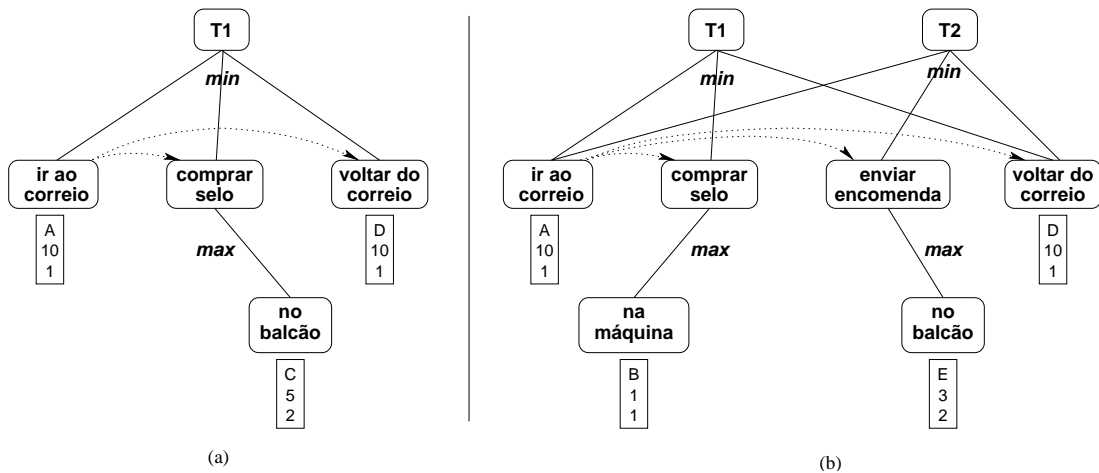


Figura 2.5: Visão objetiva de uma tarefa no modelo TÆMS (DECKER, 1998).

resolução de problemas em um dado domínio. A partir destas visões geradas, informações mais gerais (estatísticas) sobre a tarefa podem ser calculadas. Por exemplo, as visões objetivas e subjetivas das figuras 2.5 e 2.6 poderiam ter sido geradas, junto com várias outras visões, a partir de uma visão generativa. Com a avaliação das propriedades das visões geradas, descobre-se que tipo de visão é mais adequada para determinado domínio de aplicação.

Dois tipos de relações constroem uma estrutura de tarefas. O primeiro tipo é a relação de sub-tarefa que organiza as tarefas em vários níveis de abstração, formando uma árvore. As folhas desta árvore são formadas por métodos que os agentes são capazes de executar. Estes métodos também possuem vários atributos, dentre eles: a qualidade do resultado de sua execução e sua duração. Por exemplo, na figura 2.5 a tarefa *T1* (“obter selos”) é decomposta em três sub-tarefas. Junto com a relação de sub-tarefa há uma função que indica como a qualidade de uma tarefa pode ser calculada a partir de suas sub-tarefas. Dentre as várias funções propostas pelos autores, seguem algumas com propósito de exemplificá-las:

- **min:** a qualidade da tarefa é igual à menor qualidade de suas sub-tarefas (como a tarefa *T1* do exemplo). Caso uma das sub-tarefas não seja realizada, sua qualidade é igual a zero e, portanto, a qualidade da tarefa também é zero. Nestes casos, busca-se realizar todas as sub-tarefas;



Em (a) tem-se a visão de um agente que somente conhece a tarefa $T1$ e que somente pode comprar selo no balcão. Em (b), o agente conhece as duas tarefas, mas não tem como utilizar a relação de “facilita” entre elas sendo obrigado a passar na máquina para comprar selo e depois passar no balcão para despachar a encomenda.

Figura 2.6: Visão subjetiva de uma tarefa no modelo TÆMS (DECKER, 1998).

- **sum:** a qualidade da tarefa é igual a soma das qualidades de suas sub-tarefas. Nestes casos, também se busca a realização de todas as sub-tarefas, mas caso uma delas não seja realizada a qualidade da tarefa não é zero;
- **max:** a qualidade da tarefa é igual a maior qualidade de suas sub-tarefas (caso da tarefa “comprar selo” do exemplo). Nestes casos, normalmente se opta por executar somente uma das sub-tarefas⁶;
- **exactly-one:** a qualidade da tarefa é zero se mais de uma de suas sub-tarefas for executada, caso contrário é igual à qualidade da sub-tarefa realizada.

No exemplo, a tarefa “comprar selo” tem qualidade 1 se o método B for executado e qualidade 2 se o método C for executado. A tarefa $T1$ tem, no máximo, qualidade 1 se todas suas sub-tarefas forem executadas.

O segundo tipo de relação entre as tarefas, chamado de *nonlocal-effects* (NLE), permite identificar se uma tarefa altera positiva ou negativamente a qualidade de

⁶Como será visto mais à frente, não é somente o tipo de relação entre tarefa e sub-tarefa que irá determinar que tarefas serão selecionadas para execução.

outra tarefa. A NLE “ t_x habilita t_y ”, por exemplo, diz que a qualidade de t_y será zero se t_y for executada antes de t_x (comprar o selo antes de ir ao correio) e será a qualidade calculada para t_y caso contrário. A NLE “ t_x facilita t_y ”, diz que a qualidade de uma tarefa será aumentada com um fator f caso as duas tarefas sejam executadas juntas. No caso em que a NLE relaciona tarefas sendo realizadas por agentes diferentes, tem-se a necessidade de coordenação entre estes dois agentes. Percebe-se então que o cálculo da qualidade de uma tarefa depende também do escalonamento utilizado pelos agentes para realizá-las. Embora a representação do TÆMS não tenha o propósito de descrever o escalonamento (e conseqüentemente os compromissos de coordenação), esse tipo de informação pode ser gerado a partir do modelo (DECKER; LESSER, 1995; DECKER; LESSER, 1992). Resumidamente, a visão generativa gera várias visões objetivas/subjetivas para as quais vários escalonamentos podem ser gerados. A partir destes escalonamentos, é possível avaliar quais estruturas de tarefa são mais adequadas, robustas, rápidas, etc. para um determinado problema.

2.2.5 Modelo organizacional MOISE

Hannoun et al. propõem um modelo de organização, denominado MOISE (Model of Organization for multi-agent SystEms), que, além das tarefas, também considera a estrutura formada pelas relações entre os papéis dos agentes (HANNOUN et al., 2000; HANNOUN, 2002). Além das características puramente organizacionais (chamadas de nível social), o modelo proposto considera também as tarefas e responsabilidades dos agentes (nível individual — a visão subjetiva do modelo TÆMS) e o agrupamento destes agentes (nível coletivo). Para a descrição do nível individual, como nos trabalhos anteriores, o conceito de papel social é utilizado. Segundo os autores, um papel é formado por um conjunto de missões, e cada missão pelos seguintes atributos (no topo da figura 2.7 há um exemplo de definição de missão):

1. uma marcação deontica que indica se a missão é obrigatória (denotado por O) ou facultativa (denotado por P) do papel,
2. um conjunto de metas a serem alcançadas,
3. um conjunto de planos que podem ser utilizados,

Papel *candidato a estudante*, missão: $m_1 = \langle O, \{g_1\}, \{p_1\}, \{a_1\}, Any \rangle$

Metas g_1 : *estar inscrito*
 g_2 : *material de inscrição estar processado*

Planos $p_1(g_1) = a_1(r_1, r_2, r_3, r_4); [g_2]$

Ações a_1 : *arrumar o material para inscrição*

Recursos r_1 : *último diploma*
 r_2 : *histórico da graduação*
 r_3 : *foto*
 r_4 : *cópia de artigos*

Papel *secretário*, missão: $m_2 = \langle P, \{g_2\}, \{p_2\}, \{a_3, a_4\}, Any \rangle$

Metas g_2 : *material de inscrição estar processado*
 g_3 : *estudar o material do aluno*

Planos $p_2(g_2) = [g_3]; (a_3 | a_4)$

Ações a_3 : *recusar a inscrição*
 a_4 : *alterar o papel para estudante*

Figura 2.7: Descrição de uma missão no modelo MOISE (HANNOUN et al., 2000)

4. um conjunto de ações que podem ser utilizadas, e
5. um conjunto de recursos que podem ser usados.

No exemplo da figura 2.7, dois papéis são definidos, cada um com uma única missão. O agente que assume o papel de *candidato a estudante* é obrigado a realizar a missão m_1 devendo alcançar a meta g_1 e pode utilizar para este fim somente o plano p_1 , a ação a_1 e qualquer recurso (*Any*). No caso, o agente pode realizar a ação a_1 com os recursos necessários (r_1, r_2, r_3, r_4) para iniciar a execução do plano p_1 que atinge a meta de sua missão (g_1). Para que o plano seja completamente realizado, a meta g_2 deve ser alcançada, porém o agente não tem permissão de tentar realizar esta segunda meta. Neste caso o agente precisa da ajuda de outro agente (um secretário, por exemplo) para completar seu plano e atingir sua meta.

No nível social, são descritas as ligações que restringem a interação entre papéis. Há três tipos de ligações:

1. comunicação: dois papéis que possuem uma ligação de comunicação podem trocar mensagens. Esse tipo de ligação também estabelece o protocolo de comunicação que deve ser seguido caso os dois papéis venham a interagir;
2. autoridade: indica que um papel tem o direito de, por exemplo, requisitar serviços a outro;
3. conhecimento: um agente somente pode conhecer (representar mentalmente) um outro agente se um de seus papéis possuir uma ligação de conhecimento com um dos papéis do outro agente.

As ligações e os papéis sociais formam a estrutura organizacional da sociedade. Esta estrutura organizacional mais um conjunto de agentes nela alocados formam uma Entidade Organizacional (EnO), como ilustra a figura 2.8.

A vantagem de um detalhamento maior na descrição de uma estrutura organizacional é que os agentes podem raciocinar melhor sobre sua organização e identificar, por exemplo, eventuais inconsistências organizacionais.

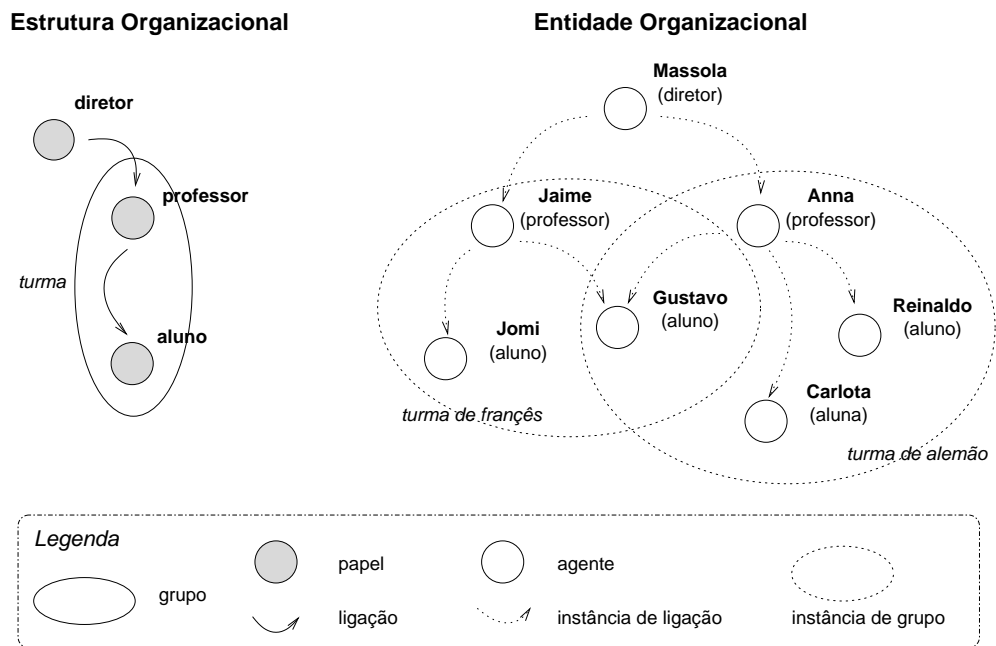
2.2.6 Organização e AOSE

Em alguns trabalhos na área de engenharia de software orientada a agentes (AOSE), a organização é vista como um instrumento de especificação do sistema, assim como ocorre, por exemplo, com o diagrama de classes da Unified Modeling Language (UML)⁷. O esforço de desenvolver uma metodologia de desenvolvimento de SMA é considerado nas seguintes propostas (somente o aspecto organizacional delas será comentado aqui):

GAIA de Wooldridge, Jennings e Kinny (1999). Esta metodologia pretende guiar o projetista de um SMA desde seu levantamento de requisitos até sua implementação. Na figura 2.9 estão ilustrados os modelos que o projetista deve desenvolver. A organização do sistema é definida por meio dos modelos de papéis e de interação que devem ser definidos na fase de análise. Cada papel, por sua vez, é formado por:

- responsabilidades (funções que o agente deve realizar para o sistema, definidas por meio de atividades e protocolos),

⁷Iglesias, Garrijo e Gonzalez (1999) apresentam um resumo destas abordagens.



A estrutura organizacional do lado esquerdo da figura é *instanciada* criando a entidade representada no lado direito. Neste exemplo, os quatro agentes que têm papel de aluno aceitam também uma ligação com um agente professor a partir do momento que passam a fazer parte de um grupo (turma de alemão, por exemplo).

Figura 2.8: Entidade organizacional no modelo MOISE (HANNOUN et al., 2000).

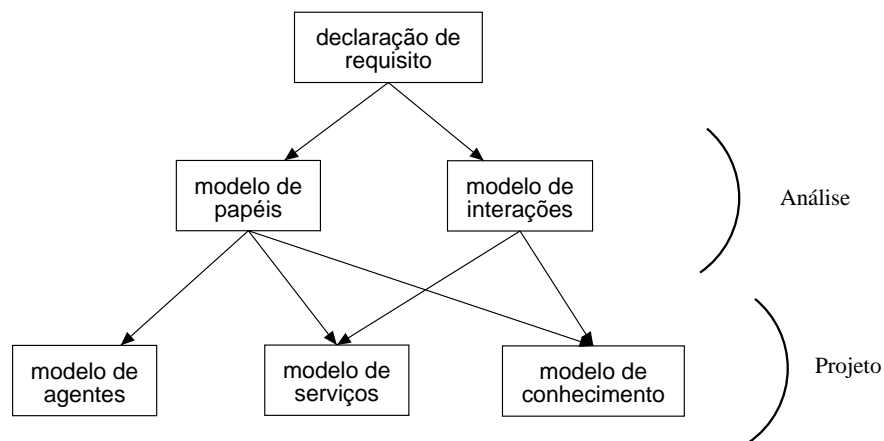


Figura 2.9: Relação entre os modelos de especificação da metodologia GAIA (WOOLDRIDGE; JENNINGS; KINNY, 1999).

- permissões para utilização de recursos,
- atividades (ações que o agente pode realizar sozinho) e
- protocolos (como o agente deve interagir com os demais).

AgentUML de Odell, Parunak e Bauer (2000). Na proposta AgentUML, ao contrário de criar uma nova notação para SMA, pretende-se estender/adequar os diagramas da UML para suportarem a especificação de SMA. A ênfase destas extensões está na especificação dos protocolos de interação e nos diagramas de colaboração entre os agentes. O aspecto organizacional, principalmente no que se refere à noção de papel, é, por enquanto, superficialmente abordado. Não há, por exemplo, um diagrama organizacional.

MESSAGE de Garijo et al. (2001), Caire et al. (2001). O MESSAGE também tem a UML por base, porém cria novos diagramas que consideram as necessidades específicas dos SMA. Existe, portanto, um diagrama organizacional onde é especificado:

- o propósito da organização (um conjunto de metas que devem ser satisfeitas pelas tarefas);
- estrutura de *workflow* (relação entre as tarefas, suas dependências e dos agentes responsáveis pelas tarefas);
- estrutura organizacional (relação dos papéis e agentes que os assumem);

- entidade de controle (define como se dá a resolução de conflitos);
- recursos da organização; e
- relações organizacionais (utilizadas para formar hierarquias entre papéis, por exemplo).

Percebe-se que o foco principal do modelo está na visão geral do sistema (a estrutura de *workflow* exemplifica este foco). Já o modelo GAIA enfatiza a visão local de cada papel.

A *utilização* que estas metodologias fazem da noção de organização difere, entretanto, da noção apresentada no início deste capítulo. Em geral, nestas metodologias a definição dos papéis serve apenas para que o projetista compreenda melhor o problema e posteriormente este papel é transformado em código. Quando o sistema está em funcionamento, não existe mais a especificação da organização *no* sistema. Portanto, o uso destas metodologias não implica em uma visão centrada na organização.

Por outro lado, tais metodologias definem um modelo organizacional e, por este motivo, podem contribuir na construção de um novo modelo organizacional voltado para a reorganização. A definição do modelo organizacional não está diretamente relacionada à utilização que lhe será dada. Um modelo por ser utilizado tanto como ferramenta de especificação quanto como componente do SMA.

2.3 Conclusões

Independentemente da posição que a organização assume em relação aos agentes que a compõem (se somente emerge de suas interações ou se existe explicitamente na sociedade), os modelos de organização desenvolvidos na área de SMA podem ser agrupados em dois grandes grupos (resumidos na tabela 2.1): aqueles que têm como foco principal os *planos globais* (ou tarefas) do SMA e aqueles que têm foco nos *papéis*.

O primeiro grupo atém-se a problemas tais como especificação de planos globais, políticas de alocação de tarefas, qualidade com que são produzidos os resultados das tarefas e principalmente na *coordenação* da execução destes planos pelos agentes (FOX, 1981; DECKER, 1996; WEISS, 1994; TAMBE, 1997; TERABE

et al., 1997; SO; DURFEE, 1996; KINNY et al., 1994). A organização é vista como um conjunto de planos globais que leva os agentes a atingirem os objetivos globais. A organização é um tipo de memória onde as melhores práticas de resolução de problemas foram armazenadas (na forma de planos ou tarefas) pelos agentes ou pelo seu projetista.

Um outro grupo de autores prioriza a *estrutura* de relações entre os agentes ou, em um nível maior de abstração, entre os *papéis* que formam a organização (FOX et al., 1998; FERBER; GUTKNECHT, 1998; PATTISON; CORKILL; LESSER, 1987; HANNOUN et al., 2000; TAMBE; PYNADATH; CHAUVAT, 2001; STONE; VELOSO, 1999). Os agentes passam a pertencer a grupos ou divisões, a terem relações de autoridade e poder, a terem certos meios de comunicação, etc. Aqui a organização contribui para a satisfação dos objetivos globais na medida que os agentes devem respeitar as responsabilidades dos papéis que assumem. Alguns dos autores deste segundo grupo propõem modelos de organização que abrangem tanto os aspectos estruturais quanto funcionais (FOX et al., 1998; HANNOUN et al., 2000; TAMBE; PYNADATH; CHAUVAT, 2001; STONE; VELOSO, 1999). O modelo MOISE, por exemplo, considera tanto os planos dos agentes (missões) quanto as relações entre eles (ligações).

Retomando a questão que motivou este capítulo (“o que é organização?”), a partir dos trabalhos citados, pode-se dizer, de forma ainda genérica e no contexto dos SMA, que

*a organização de um SMA é um conjunto de restrições ao comportamento dos agentes a fim de conduzi-los a uma finalidade comum.*⁸

Estas restrições são aceitas (Tipo OC) ou impostas (Tipo OR) aos agentes quando entram na sociedade e têm por objetivo controlar sua autonomia buscando produzir um comportamento global direcionado a uma finalidade. Tais restrições ora são descritas estruturalmente (na forma de papéis) ora funcionalmente (na forma de planos).

Considerando que a organização de um SMA tem tal objetivo, constata-se uma terceira forma, normalmente não rotulada de organização, de representar as

⁸Um artigo que descreve muito bem esta visão de organização, bem como sua importância na área, é (DIGNUM; DIGNUM, 2001).

restrições: por meio de normas sociais, leis, obrigações e direitos (CONTE; CASTELFRANCHI, 1994; CASTELFRANCHI; CONTE; PAOLUCCI, 1998; CONTE; FALCONE; SARTOR, 1999; DIGNUM; DIGNUM, 2001). O modelo MOISE, por exemplo, já inclui esse tipo de restrição quando relaciona um papel a uma missão por um operador deôntico (obrigação ou permissão).

Enfim, a partir da revisão bibliográfica realizada, identificou-se quatro tipos de *relacionamentos* entre os agentes e a sua organização (AR, AC, OR, OC) e a *descrição* desta organização pode basear-se em três componentes: estrutura de papéis, planos globais e/ou normas.

No capítulo seguinte será desenvolvido um modelo organizacional que permite a descrição destes três componentes e que pode ser utilizado nos quatro tipos de organizações.

Tabela 2.1: Comparação entre algumas propostas de modelos organizacionais

Modelo	<i>Elemento Central</i>	<i>Estrutura organizacional</i>	<i>Papel</i>	<i>Funcionamento da sociedade</i>
visões centradas nos agentes				
<i>Drogoul et al.</i> (MANTA)	agentes	emergente	não existe	determinado pela implementação dos agentes
<i>Werner</i>	papéis	formada por um conjunto de papéis	são restrições sobre o estado mental do agente	determinado indiretamente pelas restrições aos estados mentais dos agentes
visões centradas na organização				
<i>Ferber e Gutknecht</i> (AALAADIN)	papéis	é um conjunto de grupos	são funções sociais	é definido internamente nos agentes
<i>Decker</i> (TÆMS)	tarefas	tarefas e relações entre elas (sub-tarefa, habilita, facilita, ...)	restrições sobre a visão que o agente tem da tarefa	definido pelas tarefas (visões subjetiva dos agentes)
visões mistas				
<i>Fox et al.</i> (TOVE)	papéis	formada por divisões e sub-divisões, agentes, papéis, e metas sociais	é um conjunto de metas, processos, autoridades, habilidades e recursos; são relacionados hierarquicamente	determinado por metas sociais e pelos processos associados aos papéis
<i>Hannoun et al.</i> (MOISE)	papéis	formada por grupos, papéis e ligações entre papéis	é conjunto de missões; são relacionados (autoridade, comunicação, conhecimento)	dados pelas missões (obrigações ou permissões para metas, planos, ações e recursos)

3 Modelo \mathcal{MOISE}^+

Enquanto o capítulo anterior permite estabelecer *o que é* uma organização, este capítulo detalha *como* a organização de um SMA pode contribuir na manutenção e busca da sua finalidade. Este detalhamento é feito por meio de uma proposta de um modelo organizacional, denominado \mathcal{MOISE}^+ . Este modelo estabelece precisamente quais os componentes que formam uma organização e como estes podem contribuir para a finalidade do SMA. Além desta função ontológica, o modelo \mathcal{MOISE}^+ foi desenvolvido para auxiliar o processo de reorganização. Apresenta, portanto, características que suportam tanto a avaliação quanto o projeto de novas organizações.

O modelo \mathcal{MOISE}^+ é fortemente baseado no modelo MOISE (seção 2.2.5), que além de apresentar uma visão centrada na organização considera claramente as três formas de representar restrições organizacionais (papéis, planos e normas). Como consequência da visão centrada na organização, o \mathcal{MOISE}^+ possui duas noções centrais: uma *especificação organizacional* (seções 3.2, 3.3 e 3.4) que um grupo de agentes adota para formar uma *entidade organizacional* (seção 3.5), cuja ação se destina a atingir uma finalidade.

3.1 Introdução

Na construção do modelo \mathcal{MOISE}^+ , optou-se por conceber a organização de um SMA como possuindo as três dimensões identificadas na seção anterior (seção 2.3): a estrutura (papéis), o funcionamento (planos globais) e as normas (obrigações) da organização. O aspecto estrutural atém-se aos componentes elementares da organização (papéis) e como estão relacionados (ligações entre papéis, grupos de papéis, hierarquias). O aspecto funcional especifica como os objetivos globais podem ser atingidos (planos globais, missões e metas). Por fim, o aspecto deontico

liga os dois anteriores, indicando quais as responsabilidades dos papéis nos planos globais.

A relação entre estas três dimensões e a forma como contribuem para a finalidade do SMA são ilustradas na figura 3.1. Nesta figura, B representa o espaço de comportamentos possíveis para o sistema e P os comportamentos que conduzem à finalidade do SMA ($P \subset B$). Por *comportamento* de um sistema entende-se suas ações no ambiente, dadas as percepções que teve do mesmo ambiente. Sendo O o conjunto das percepções possíveis e A o conjunto das ações possíveis para o sistema, um comportamento b (um ponto no espaço de comportamentos) é representado por um mapeamento entre uma seqüência de observações do ambiente (denotada por $(o_1, o_2, \dots) \in O^*$) e uma ação do sistema (denotada por $\alpha \in A$) onde $b : O^* \rightarrow A$.

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 ($S \subset B$), mais precisamente, a cada papel é relacionado 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, encontrar P dentro de $E \cap S$ pode ser um problema para os agentes, nos casos onde esta intersecção ainda é um conjunto muito grande de comportamentos possíveis. Para ajudá-los nesta 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 $E \cap S \cap P$. Neste contexto, a especificação deôntica indica que papéis (em S) devem utilizar os comportamentos definidos em F .

Normalmente não se deseja especificar uma organização onde os comportamentos possíveis são exatamente aqueles que levam à finalidade ($S = P$). Como P pode eventualmente mudar e os comportamentos que levam à finalidade passarem a ser P' , se inicialmente $S = P$ e a finalidade mudar, a organização pode deixar de ser eficaz ($S \cap P' = \emptyset$). Em outras palavras, se S é muito pequeno, o SMA pode ter problemas de adaptação a mudanças, já que a autonomia dos agentes pode ter sido extinta. Por outro lado, se o conjunto S é muito grande, a organização não é eficiente, já que esta não restringe a autonomia dos agentes. Surge aqui, de forma mais clara, o problema que motivou essa tese (já apresen-

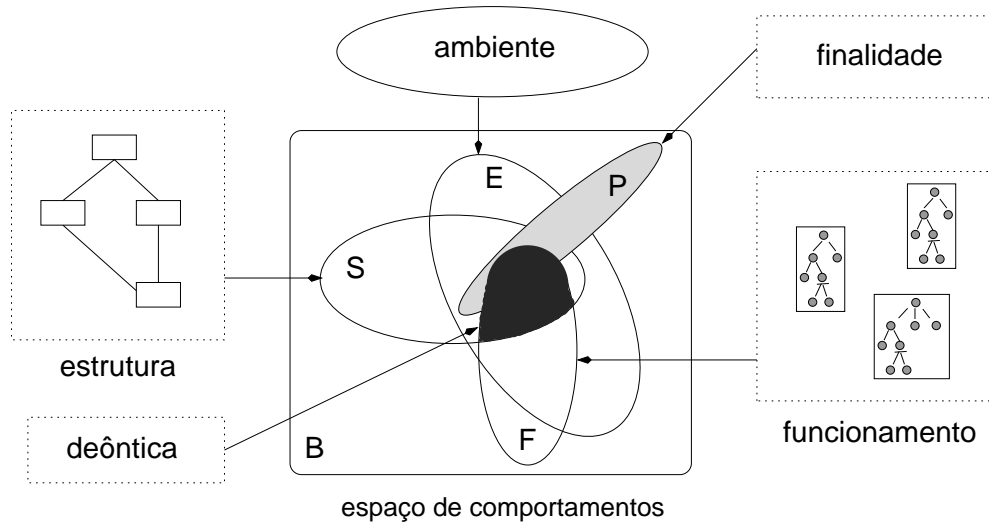


Figura 3.1: Contribuição da organização para a finalidade do SMA.

tado na introdução): como conciliar a organização com a autonomia dos agentes, isto é, como especificar uma boa organização. A visão apresentada na figura 3.1 permite considerar o problema de encontrar uma boa organização (um bom tamanho para $S \cap F$) como um problema de busca no espaço de comportamentos (B).¹ O sucesso na manutenção da finalidade em uma sociedade depende da sua organização (que determina os comportamentos permitidos $S \cap F$) e do ambiente. Como os agentes não têm controle total sobre o ambiente, uma forma de manter a finalidade da sociedade numa eventual mudança de ambiente é alterar sua organização nos aspectos estruturais, funcionais e/ou normativos.

Dada esta visão geral do modelo e como ele pode contribuir para a finalidade de um SMA, o restante do capítulo o define precisamente.

Definição 3.1 (Especificação Organizacional) O conjunto de todas as especificações organizacionais (\mathcal{OS}) possíveis é formado por tuplas de aridade três:

$$\mathcal{OS} =_{\text{def}} \mathcal{SS} \times \mathcal{FS} \times \mathcal{DS} \quad (3.1)$$

¹Considerar o problema de alcançar um objetivo social como um problema de busca é uma estratégia proposta por Durfee e Montgomery (1991) aplicada ao problema de coordenação entre agentes. Eles definiram o espaço de busca em seis dimensões: quem, (faz) o que, quanto, onde, como e por que. A busca consiste em encontrar um sextupla que resolva o problema dado. Contudo, Durfee e Montgomery (1991) consideram somente aspectos funcionais (o que, como e por que) e de escalonamento (quem, quando e onde) nesta busca, os aspectos estruturais não estão incluídos ou não estão explícitos.

onde: \mathcal{SS} é o conjunto das especificações estruturais; \mathcal{FS} é o conjunto das especificações funcionais; e \mathcal{DS} é o conjunto das especificações deônticas. \square

Cada uma destas três especificações é detalhada nas seções seguintes onde utilizou-se a teoria dos conjuntos e a lógica de primeira ordem para sua formalização.

3.2 Especificação estrutural

No modelo \mathcal{MOISE}^+ , como no \mathcal{MOISE} , três conceitos principais, denominados papéis, relações entre papéis e grupos, são utilizados para construir, respectivamente, os níveis individual, social e coletivo de uma Especificação Estrutural (EE).

3.2.1 Nível individual: papéis

Apesar do conceito de papel não ser consensual, a maioria dos autores concordam que um papel é um conjunto de *restrições comportamentais* que um agente aceita ao entrar em um grupo (HÜBNER, 1995; FERBER; GUTKNECHT, 1998; HANNOUN et al., 2000; GLASER; MORIGNOT, 1997; CAVEDON; SONENBERG, 1998). Mais precisamente, segundo (CASTELFRANCHI, 1996), há dois tipos de restrições comportamentais:

- (i) aquelas que um papel confere a um agente em relação a outro agente (por exemplo, considerando os papéis de general e soldado, os agentes que assumem estes papéis têm uma relação de autoridade definida pelos papéis que possuem) e
- (ii) obrigações para com tarefas comuns (por exemplo, os pais devem educar os filhos).

Há portanto um aspecto estrutural dos papéis (i), que será visto nesta seção, e outro funcional (ii), que será desenvolvido na seção 3.3.

No modelo de organização aqui proposto, o conceito de papel tem função primordial por ser o elo de ligação entre o agente e a organização. Como já foi dito, esta ligação tem tanto uma dimensão estrutural, que é estabelecida quanto um

agente assume um papel, quanto funcional, que é estabelecida quando um agente se compromete com uma missão. Na estrutura, as ligações que um papel possui com outros papéis são definidas com o objetivo de restringir o comportamento de um papel² em relação a outros papéis. Na dimensão funcional, a restrição ocorre por meio do conjunto de objetivos que o papel deve ou pode ter.

Na formalização do modelo MOISE^+ , um papel é simplesmente um identificador sobre o qual serão definidas relações com os outros elementos da organização. A compreensão de um papel organizacional, ou seja, o conjunto de restrições comportamentais que representa, é obtido a partir das relações que tal papel tem na Especificação Organizacional (EO).

Será considerado também um tipo especial de papel, chamado de papel abstrato, que nenhum agente pode assumir. Tais papéis existem somente para simplificar a especificação de uma organização.

Definição 3.2 (*Papel*) Um papel ρ é um identificador que pertence ao conjunto de identificadores de papéis \mathcal{R}_{ss} da EE ($\rho \in \mathcal{R}_{ss}$). \square

Definição 3.3 (*Papel abstrato*) Um papel abstrato é um papel que não pode ser assumido pelos agentes. O conjunto de todos os papéis abstratos formam o conjunto \mathcal{R}_{abs} sendo que a seguinte propriedade deve ser satisfeita

$$\mathcal{R}_{abs} \subset \mathcal{R}_{ss} \quad (3.2)$$

\square

Para simplificar o processo de especificação dos papéis de uma organização, uma primeira relação que se estabelece entre os papéis é a relação de herança (FOX et al., 1998; ESTEVA et al., 2001). Se um papel ρ' é herdeiro do papel ρ (denotado por $\rho \sqsubset \rho'$), então ρ' tem algumas das propriedades de ρ mais aquelas que lhe são específicas³. Em analogia aos termos de Orientação a Objetos (OO), diz-se neste caso que ρ' é uma *especialização*, ou sub-papel, de ρ e que ρ é uma *generalização*, ou super-papel, de ρ' . Um papel pode também ser uma especialização de mais de um papel, isto é, pode receber propriedades de mais de um papel. Por exemplo, o

²Na verdade, um papel não tem comportamento, trata-se do comportamento do agente que assume tal papel. A fim de simplificar o texto, na seqüência, o termo papel será utilizado também como sinônimo de “o conjunto dos agentes que assumem tal papel”.

³No texto que segue, serão detalhadas quais são as propriedades sujeitas à herança.

papel de aluno de pós-graduação pode ser definido como herdando propriedades do papel mais geral de aluno e de pesquisador ($\rho_{aluno} \sqsubset \rho_{alunopos}$, $\rho_{pesquisador} \sqsubset \rho_{alunopos}$). Continuando nesta “inspiração” baseada na OO, há um super-papel comum a todos os papéis, tal papel é chamado de ρ_{soc} .

Definição 3.4 (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):

$$\rho \sqsubset \rho' \wedge \rho' \sqsubset \rho \Rightarrow \rho = \rho' \quad (3.3)$$

$$\rho \sqsubset \rho' \wedge \rho' \sqsubset \rho'' \Rightarrow \rho \sqsubset \rho'' \quad (3.4)$$

□

Adicionalmente, existe um papel abstrato ρ_{soc} que é a raiz da árvore de papéis, como mostra a definição seguinte.

Definição 3.5 (Papel social) Dado o papel abstrato ρ_{soc} , acrescentam-se as seguintes propriedades ao conjunto de papéis e à relação de herança:

$$\rho_{soc} \in \mathcal{R}_{ss} \quad (3.5)$$

$$\forall \rho \in (\mathcal{R}_{ss} - \{\rho_{soc}\}) \bullet \rho_{soc} \sqsubset \rho \quad (3.6)$$

$$\nexists \rho \in \mathcal{R}_{ss} \bullet \rho \sqsubset \rho_{soc} \quad (3.7)$$

□

3.2.2 Nível social: ligações e compatibilidades

Enquanto a relação de herança não tem efeito direto no comportamento dos agentes, no nível social os papéis estão relacionados a outros papéis representando restrições impostas às interações de um papel com os outros.

Um primeiro tipo de restrição entre papéis são as ligações (HANNOUN et al., 2000). Uma ligação é representada pelo predicado $link(\rho_s, \rho_d, t)$ onde ρ_s é o

papel de origem, ρ_d é o papel destino e t é o tipo da ligação, sendo possível três tipos distintos. No caso do tipo t da ligação ser *acq* (conhecimento), os agentes que assumem o papel de origem ρ_s têm permissão de representar (conhecer) os agentes que assumem o papel destino ρ_d da ligação⁴. No caso de uma ligação de comunicação ($t = com$), os agentes ρ_s têm permissão para se comunicar com os agentes ρ_d . Na ligação de autoridade ($t = aut$), os agentes ρ_s têm autoridade sobre os agentes ρ_d .

Definição 3.6 (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
- $\rho_s \in \mathcal{R}_{ss} \wedge \rho_d \in \mathcal{R}_{ss} \wedge t \in \mathcal{LT}$

onde $\mathcal{LT} = \{acq, com, aut\}$. As fórmulas seguintes são verdadeiras para o predicado de ligação:

$$link(\rho_s, \rho_d, t) \wedge \rho_s \sqsubset \rho'_s \Rightarrow link(\rho'_s, \rho_d, t) \quad (3.8)$$

$$link(\rho_s, \rho_d, t) \wedge \rho_d \sqsubset \rho'_d \Rightarrow link(\rho_s, \rho'_d, t) \quad (3.9)$$

$$link(\rho_s, \rho_d, aut) \Rightarrow link(\rho_s, \rho_d, com) \quad (3.10)$$

$$link(\rho_s, \rho_d, com) \Rightarrow link(\rho_s, \rho_d, acq) \quad (3.11)$$

□

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.

Por exemplo, se um docente tem autoridade sobre um aluno $link(\rho_{docente}, \rho_{aluno}, aut)$, então, pela equação (3.10), o docente também pode se comunicar com o aluno e, pela equação (3.11), pode conhecê-lo. No mesmo exemplo, se o papel de aluno tem o papel de pós-graduando como uma especialização ($\rho_{aluno} \sqsubset \rho_{alunopos}$), então o docente, pela equação (3.9), também tem autoridade sobre o pós-graduando.

⁴O modelo \mathcal{MOISE}^+ não detalha o que pode ser conhecido do outro agente. Na arquitetura proposta no capítulo 4, um agente pode conhecer o nome dos demais se possuir uma ligação de comunicação para eles.

Enquanto as ligações restringem o comportamento dos agentes depois que eles assumem um papel, a relação de compatibilidade restringe os papéis que um agente pode desempenhar em função dos papéis correntes que já desempenha. Por exemplo, um agente não pode assumir o papel de aluno na turma onde é professor. A relação de compatibilidade $\rho_a \bowtie \rho_b$ determina que um agente com papel ρ_a pode assumir o papel ρ_b . Se não for explicitamente especificado que dois papéis são compatíveis, assume-se que eles não o são.

Definição 3.7 (Compatibilidade) O predicado de $\rho_a \bowtie \rho_b$ é verdade se

- um agente com o papel ρ_a pode também assumir 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 herança):

$$\rho \bowtie \rho \quad (3.12)$$

$$\rho \bowtie \rho' \wedge \rho' \bowtie \rho'' \Rightarrow \rho \bowtie \rho'' \quad (3.13)$$

$$\rho_a \bowtie \rho_b \wedge \rho_a \neq \rho_b \wedge \rho_a \sqsubset \rho' \Rightarrow \rho' \bowtie \rho_b \quad (3.14)$$

$$\rho \sqsubset \rho' \Rightarrow \rho' \bowtie \rho \quad (3.15)$$

□

A última implicação pode ser lida como “os papéis são compatíveis com seus super-papéis”. Por exemplo, se o papel de aluno tem aluno de pós-graduação como sub-papel, então, por definição, todo agente com papel de aluno de pós-graduação pode também assumir o papel de aluno ($\rho_{aluno} \sqsubset \rho_{alunopos} \Rightarrow \rho_{alunopos} \bowtie \rho_{aluno}$), mas o contrário não é válido por definição (a relação de compatibilidade não é simétrica). Caso se pretenda estabelecer que todo aluno pode ser aluno de pós-graduação, esta compatibilidade ($\rho_{aluno} \bowtie \rho_{alunopos}$) deve ser explicitamente acrescentada à especificação.

3.2.3 Nível coletivo: grupos

Os papéis somente são assumidos no nível coletivo, isto é, dentro de um *grupo*. Intuitivamente, um grupo representa um conjunto de agentes com afinidades mai-

ores e objetivos mais próximos (FERBER; GUTKNECHT, 1998; HANNOUN et al., 2000). Um departamento de uma empresa ou os jogadores da defesa em um time exemplificam o conceito de grupo dentro de uma organização. Um grupo diferencia-se de uma sociedade justamente por agir como se fosse um único agente cujo comportamento é direcionado a objetivos comuns. Por exemplo, os motoristas em uma cidade conduzem seus carros coordenadamente, mas não formam um grupo. Já um comboio de carros apresenta uma meta comum e, portanto, caracteriza melhor a noção de grupo (LEVESQUE; COHEN; NUNES, 1990). Além da similaridade com o mundo real, a divisão de agentes em grupos facilita o desenvolvimento e o projeto de uma organização. O projetista pode se ater à especificação de um grupo menor de agentes e usar estes grupos na definição de várias estruturas organizacionais diferentes.

Como já foi apresentado, no modelo \mathcal{MOISE}^+ distinguem-se duas noções: a especificação da organização (EO), que é estática, e a criação de entidades que seguem esta especificação (EnO). Por exemplo, um papel (uma especificação) que um agente (uma entidade) assume. De forma análoga, há duas noções de grupo: a especificação de um tipo de grupo e um grupo de agentes que adota tal especificação. Será utilizado o termo “especificação de grupo” para indicar um *tipo* de grupo (o aspecto estático que está sendo descrito nesta seção) e o termo “grupo” para significar um grupo instanciado a partir de tal especificação (mais detalhes da criação de grupos são apresentados no capítulo 4).

Uma especificação de grupo consiste

- dos papéis que podem ser assumidos no grupo (por exemplo, em um grupo do tipo turma, somente os papéis de aluno e professor podem ser assumidos),
- dos sub-grupos que podem ser criados dentro do grupo (por exemplo, dentro do grupo escola, e somente dentro de escola, pode-se criar grupos do tipo turma),
- de ligações que são válidas para os agentes do grupo (por exemplo, um professor tem autoridade sobre os alunos das turmas onde é professor),
- de compatibilidades que são válidas para os agentes do grupo (por exemplo, na mesma turma, um professor não pode ser aluno), e

- de cardinalidades que determinam a boa formação do grupo (por exemplo, uma turma só pode ter um professor).

Definição 3.8 (Especificação de Grupo) Sendo \mathcal{GT} o conjunto de todas as especificações de grupo da EE, uma especificação de grupo $gt \in \mathcal{GT}$ é representada pela tupla

$$gt = (\mathcal{R}, \mathcal{SG}, \mathcal{L}^{intra}, \mathcal{L}^{inter}, \mathcal{C}^{intra}, \mathcal{C}^{inter}, np, ng) \quad (3.16)$$

onde:

- \mathcal{R} o conjunto de papéis que podem ser assumidos em grupos criados a partir da especificação de grupo gt

$$\mathcal{R} \subseteq \mathcal{R}_{ss} \quad (3.17)$$

- \mathcal{SG} o conjunto de sub-grupos possíveis no grupo gt

$$\mathcal{SG} \subseteq \mathcal{GT} \quad (3.18)$$

- \mathcal{L}^{intra} e \mathcal{L}^{inter} os conjuntos das ligações internas ou externas ao grupo gt ;
- \mathcal{C}^{intra} e \mathcal{C}^{inter} os conjuntos das compatibilidades internas e externas ao grupo gt ;
- $np : \mathcal{R}_{ss} \rightarrow \mathbb{N} \times \mathbb{N}$ um mapeamento parcial para a cardinalidade de papéis que indica respectivamente um valor mínimo e máximo para cada papel;
- $ng : \mathcal{SG} \rightarrow \mathbb{N} \times \mathbb{N}$ um mapeamento parcial para a cardinalidade de sub-grupos que indica respectivamente um valor mínimo e máximo para cada sub-grupo.

□

Como podem existir várias especificações de grupo em uma mesma EO, também podem existir vários conjuntos de papéis, sub-grupos, ligações, etc. Quando for necessário diferenciá-los no texto, será pré-fixada a identificação da especificação de grupo. Por exemplo, o conjunto de papéis de um grupo escola ($escola \in \mathcal{GT}$) será denotado por $escola.\mathcal{R}$, e o conjunto de papéis de um grupo turma ($turma \in \mathcal{GT}$) será $turma.\mathcal{R}$.

3.2.3.1 Ligações internas e externas de um grupo

No nível coletivo, as ligações passam a possuir um escopo: internas ou externas ao grupo. Portanto, um grupo possui ligações intra-grupo \mathcal{L}^{intra} e ligações inter-grupo \mathcal{L}^{inter} . Uma ligação intra-grupo $l \in \mathcal{L}^{intra}$ estabelece que todos os agentes que assumem o papel origem de l em um grupo gr ⁵ estão ligados por l a todos os agentes que assumem o papel destino de l no *mesmo* grupo gr . As ligações inter-grupo estendem o escopo da ligação estabelecendo que todos os agentes que assumem o papel origem de l estão ligados por l a todos os agentes que assumem o papel destino de l não importando em que grupos estes agentes assumem os papéis destino e origem da ligação. Portanto, toda ligação inter-grupo implica em uma ligação intra-grupo:

$$l \in \mathcal{L}^{inter} \Rightarrow l \in \mathcal{L}^{intra} \quad (3.19)$$

Por exemplo, se $link(\rho_{aluno}, \rho_{professor}, com) \in \mathcal{L}^{inter}$, então qualquer agente com papel de aluno pode conversar com qualquer agente com papel de professor mesmo que os dois agentes estejam em grupos diferentes. No caso de $link(\rho_{professor}, \rho_{aluno}, aut) \in \mathcal{L}^{intra}$, um agente professor somente tem autoridade sobre os alunos do grupo (turma) onde desempenha o papel de professor.

As ligações definidas em uma especificação de grupo gt são válidas também nos seus sub-grupos:

$$l \in gt.\mathcal{L}^{intra} \Rightarrow \forall sg \in gt.SG \bullet l \in sg.\mathcal{L}^{intra} \quad (3.20)$$

$$l \in gt.\mathcal{L}^{inter} \Rightarrow \forall sg \in gt.SG \bullet l \in sg.\mathcal{L}^{inter} \quad (3.21)$$

3.2.3.2 Compatibilidades internas e externas de um grupo

Assim como as ligações, as compatibilidades também passam a possuir um escopo quando incluídas em um grupo. Se $\rho_a \bowtie \rho_b \in \mathcal{C}^{intra}$ (denota-se também $\rho_a \bowtie^{intra} \rho_b$) então um agente ρ_a em um grupo gr pode assumir também o papel ρ_b no *mesmo* grupo gr . Caso $\rho_a \bowtie \rho_b \in \mathcal{C}^{inter}$, então um agente ρ_a em um grupo gr_1 pode assumir também o papel ρ_b somente em outro grupo gr_2 ($gr_1 \neq gr_2$). Por exemplo, para especificar que um agente não pode assumir os papéis de professor

⁵ gr é um grupo criado a partir de uma especificação de grupo gt , tal que $gt \in \mathcal{GT}$ e $l \in gt.\mathcal{L}^{intra}$.

e aluno no mesmo grupo (turma), mas pode assumí-los em grupos diferentes, adiciona-se a compatibilidade $\rho_{professor} \bowtie \rho_{aluno}$ no conjunto de compatibilidades inter-grupo e não se adiciona esta compatibilidade no conjunto de compatibilidades intra-grupo.

As compatibilidades em uma especificação de grupo gt são válidas também nos seus sub-grupos:

$$\rho_a \bowtie \rho_b \in gt.\mathcal{C}^{intra} \Rightarrow \forall sg \in gt.\mathcal{SG} \bullet \rho_a \bowtie \rho_b \in sg.\mathcal{C}^{intra} \quad (3.22)$$

$$\rho_a \bowtie \rho_b \in gt.\mathcal{C}^{inter} \Rightarrow \forall sg \in gt.\mathcal{SG} \bullet \rho_a \bowtie \rho_b \in sg.\mathcal{C}^{inter} \quad (3.23)$$

3.2.3.3 Cardinalidades definidas em um grupo

As funções np e ng de uma especificação de grupo estabelecem uma *cardinalidade* para papéis e sub-grupos, respectivamente. A especificação destas duas funções define limites mínimos e máximos para estes elementos do grupo. Deste modo, tal cardinalidade permite associar o atributo “bem formado” a um grupo. Por exemplo, na especificação de grupo turma, pode-se dizer que $np(\rho_{professor}) = (1, 1)$ e $np(\rho_{aluno}) = (5, 30)$, portanto uma turma é bem formada se um, e somente um, agente assume o papel de professor e se cinco a trinta agentes assumem o papel de aluno. Caso contrário, o grupo não é bem formado. Como as funções np e ng são parciais, nos casos onde não é definido um mapeamento, assume-se para a imagem o par $(0, \infty)$.

Dada uma especificação de grupo gt , o domínio da função $gt.np : \mathcal{R}_{ss} \mapsto \mathbb{N} \times \mathbb{N}$ não é o conjunto de papéis do grupo ($gt.\mathcal{R}$), mas o conjunto de todos os papéis da EO (\mathcal{R}_{ss}). No caso da função mapear um valor para um papel que não esteja em $gt.\mathcal{R}$, significa que a cardinalidade será aplicada a todos os sub-grupos de gt . Por exemplo, supondo a existência de um grupo escola que tem o sub-grupo turma, onde se estabelece que $escola.np(\rho_{aluno}) = (0, 300)$ e o papel de aluno não existe no grupo escola ($\rho_{aluno} \notin escola.\mathcal{R}$), mas existe em seus sub-grupos ($\rho_{aluno} \in turma.\mathcal{R}$), tem-se então que esta cardinalidade se aplica a todas as turmas criadas dentro do grupo escola (a escola pode ter no máximo 300 agentes assumindo o papel de aluno, não importando em que grupos).

Dada a definição de especificação de grupo, outras definições são elaboradas em seguida.

Definição 3.9 (Especificação de grupo raiz) 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 raiz:

$$isRootGr(gt) \Leftarrow \nexists g \in \mathcal{GT} \bullet gt \in g.SG \quad (3.24)$$

□

Retomando a definição de especificação organizacional (EO) iniciada na página 38 (3.1), o primeiro elemento de uma EO, a especificação estrutural, é definido a seguir.

Definição 3.10 (Especificação Estrutural) Sendo SS o conjunto de todas as especificações estruturais, uma EE ss é representada por uma tupla

$$ss = (\mathcal{RG}, \mathcal{R}_{ss}, \sqsubset) \quad (3.25)$$

onde:

- \mathcal{RG} é o conjunto de especificações de grupos raízes de ss

$$\mathcal{RG} \subseteq \mathcal{GT} \quad (3.26)$$

$$\forall gt \in \mathcal{RG} \bullet isRootGr(gt) \quad (3.27)$$

- \mathcal{R}_{ss} é o conjunto de todos os papéis da EE; e
- \sqsubset é a relação de herança sobre os papéis de \mathcal{R}_{ss} .

As ligações, compatibilidades e cardinalidades entre os papéis de ss são definidas no escopo dos grupos e, portanto, estão incluídas em \mathcal{RG} (conforme 3.8).

O conjunto de papéis abstratos de uma EE é calculado a partir do conjunto de todos os papéis menos os papéis que podem ser assumidos em algum grupo:

$$\mathcal{R}_{abs} = \mathcal{R}_{ss} - \bigcup_{\forall gt \in \mathcal{GT}} gt.\mathcal{R} \quad (3.28)$$

□

Exemplo 3.1 Este exemplo especifica o caso de uma escola inicialmente descrita

$gt_{escola} = \langle$	
$\{\},$	% \mathcal{R}
$\{gt_{turma}, gt_{corpodoc}\},$	% \mathcal{SG}
$\{\},$	% \mathcal{L}^{intra}
$\{link(\rho_{docente}, \rho_{docente}, com)\},$	% \mathcal{L}^{inter}
$\{\},$	% \mathcal{C}^{intra}
$\{\},$	% \mathcal{C}^{inter}
$\{\rho_{aluno} \mapsto (0, 300)\},$	% np
$\{gt_{corpodoc} \mapsto (1, 1)\}$	% ng
$gt_{corpodoc} = \langle$	
$\{\rho_{diretor}, \rho_{professor}\},$	% \mathcal{R}
$\{\},$	% \mathcal{SG}
$\{link(\rho_{diretor}, \rho_{professor}, aut)\},$	% \mathcal{L}^{intra}
$\{\},$	% \mathcal{L}^{inter}
$\{\rho_{professor} \bowtie \rho_{diretor}\},$	% \mathcal{C}^{intra}
$\{\},$	% \mathcal{C}^{inter}
$\{\rho_{diretor} \mapsto (1, 1)\},$	% np
$\{\}\rangle$	% ng
$gt_{turma} = \langle$	
$\{\rho_{professor}, \rho_{aluno}\},$	% \mathcal{R}
$\{\},$	% \mathcal{SG}
$\{link(\rho_{docente}, \rho_{aluno}, aut), link(\rho_{aluno}, \rho_{professor}, com)\},$	% \mathcal{L}^{intra}
$\{link(\rho_{aluno}, \rho_{aluno}, com), link(\rho_{aluno}, \rho_{docente}, acq)\},$	% \mathcal{L}^{inter}
$\{\},$	% \mathcal{C}^{intra}
$\{\rho_{aluno} \bowtie \rho_{professor}, \rho_{professor} \bowtie \rho_{aluno}\},$	% \mathcal{C}^{inter}
$\{\rho_{professor} \mapsto (1, 1), \rho_{aluno} \mapsto (5, 30)\},$	% np
$\{\}\rangle$	% ng

A primeira especificação de grupo (gt_{escola}) não possui papéis que possam ser assumidos: os papéis somente podem ser assumidos nos sub-grupos, corpo docente e turma. Qualquer docente (independente de grupo, isto é, inter-grupo) pode se comunicar com outro docente. Não há compatibilidades neste grupo e uma instância do grupo escola é bem formada se possuir uma instância do sub-grupo corpo docente e, no máximo, 300 alunos em seus sub-grupos ($\rho_{aluno} \mapsto (0, 300)$).

Esta última restrição é caracterizada desta forma porque o papel de aluno não pode ser assumido no grupo escola (somente nos seus sub-grupos). Como não há cardinalidade para o sub-grupo turma, a escola pode ter qualquer número de turmas. Percebe-se também a utilidade dos papéis abstratos na construção de especificações. Neste exemplo, através de uma única ligação no papel de docente pode-se dar direito de comunicação tanto ao diretor quanto ao professor.

No grupo corpo docente, os papéis possíveis são de diretor e professor. O diretor tem autoridade sobre os professores dentro deste grupo (pois a ligação de autoridade entre os dois é intra-grupo). Um professor pode assumir o papel de diretor (compatibilidade) e deve existir um diretor no grupo. Como ilustra este exemplo, um papel pode pertencer a mais de uma especificação de grupo (por exemplo, professor). Dois professores do corpo docente podem se comunicar pois a ligação $link(\rho_{docente}, \rho_{docente}, com)$, definida no grupo escola, é válida também no grupo corpo docente por ser um sub-grupo de escola.

Por fim, em uma turma dois papéis podem ser assumidos, professor e aluno. Qualquer docente tem autoridade sobre um aluno desde que estejam no mesmo grupo (ligação intra-grupo). Os alunos podem se comunicar com o professor da turma onde são alunos (ligação intra-grupo) e podem conhecer todos os docentes (independente de grupo). Podem ainda se comunicar com qualquer outro aluno (independente de grupo). Quanto à compatibilidade, um professor pode ser aluno, mas somente em turmas diferentes (compatibilidade intra-grupo) e um aluno pode ser professor em outra turma. Uma turma é bem formada se possui um professor e de cinco a trinta alunos. Como turma é sub-grupo de escola, possui todas as ligações definidas em escola. Portanto, um professor tem autoridade sobre os alunos da turma onde é professor.

Todas as ligações que podem ser inferidas não são representadas nem na notação textual nem na notação gráfica de uma estrutura organizacional. Para exemplificar as implicações definidas nas fórmulas, tem-se que o papel de professor herda as seguintes ligações do papel de docente:

- ligação inter-grupo de comunicação com outros professores.

Prova:

- | | | |
|----|--|---|
| 1. | $link(\rho_{docente}, \rho_{docente}, com)$ | da especificação $gt_{turma} \cdot \mathcal{L}^{inter}$ |
| 2. | $\rho_{docente} \sqsubset \rho_{professor}$ | da especificação ss_{escola} |
| 3. | $(link(\rho_s, \rho_d, t) \wedge \rho_s \sqsubset \rho'_s) \Rightarrow link(\rho'_s, \rho_d, t)$ | equação (3.8) |
| 4. | $link(\rho_{professor}, \rho_{docente}, com)$
$\in gt_{turma} \cdot \mathcal{L}^{inter}$ | de 1, 2 e 3 e das substituições
$[\rho'_s / \rho_{professor}, \rho_s / \rho_{docente},$
$\rho_d / \rho_{docente}, t / com]$ |
| 5. | $(link(\rho_s, \rho_d, t) \wedge \rho_d \sqsubset \rho'_d) \Rightarrow link(\rho_s, \rho'_d, t)$ | equação (3.9) |
| 6. | $link(\rho_{professor}, \rho_{professor}, com)$
$\in gt_{turma} \cdot \mathcal{L}^{inter}$ | de 4, 2 e 5 e das substituições
$[\rho_s / \rho_{professor}, \rho'_d / \rho_{professor},$
$\rho_d / \rho_{docente}, t / com]$ |
- ligação intra-grupo de comunicação com outros professores.

Prova:

1. $link(\rho_{professor}, \rho_{professor}, com)$ da conclusão da última prova
 $\in gt_{turma} \cdot \mathcal{L}^{inter}$
 2. $l \in \mathcal{L}^{inter} \Rightarrow l \in \mathcal{L}^{intra}$ equação (3.19)
 3. $link(\rho_{professor}, \rho_{professor}, com)$ de 1 e 2
 $\in gt_{turma} \cdot \mathcal{L}^{intra}$
- ligação inter-grupo de comunicação com o diretor (com prova semelhante às anteriores);
 - ligação intra-grupo de autoridade para com os alunos; e
 - pode ser conhecido por qualquer aluno.

O papel de professor, pela equação (3.15), é compatível com o papel de docente.

□

Exemplo 3.2 Ainda no exemplo da escola, a fim de definir uma EE para o processo de seleção e ingresso em um curso de pós-graduação, propõe-se a definição de um novo grupo: comissão de seleção. Os papéis deste grupo e suas ligações estão representados na figura 3.3. O grupo escola poderia passar então a ter três sub-grupos:

$gt_{escola2} = \langle$	
$\{\},$	% \mathcal{R}
$\{gt_{turma}, gt_{corporadoc}, gt_{selecao}\},$	% \mathcal{SG}
$\{\},$	% \mathcal{L}^{intra}
$\{link(\rho_{docente}, \rho_{docente}, com)\},$	% \mathcal{L}^{inter}
$\{\},$	% \mathcal{C}^{intra}
$\{\},$	% \mathcal{C}^{inter}
$\{\rho_{aluno} \mapsto (0, 300)\},$	% np
$\{gt_{corporadoc} \mapsto (1, 1), gt_{selecao} \mapsto (0, 1)\}\rangle$	% ng
$gt_{selecao} = \langle$	
$\{\rho_{membro}, \rho_{presidente}, \rho_{secretario}, \rho_{candidato}\},$	% \mathcal{R}
$\{\},$	% \mathcal{SG}
$\{link(\rho_{membro}, \rho_{candidato}, com), link(\rho_{candidato}, \rho_{membro}, com),$	% \mathcal{L}^{intra}
$link(\rho_{presidente}, \rho_{secretario}, aut), link(\rho_{secretario}, \rho_{candidato}, com),$	
$link(\rho_{candidato}, \rho_{secretario}, com)\},$	
$\{link(\rho_{funcionario}, \rho_{docente}, com), link(\rho_{docente}, \rho_{funcionario}, com),$	% \mathcal{L}^{inter}
$link(\rho_{funcionario}, \rho_{funcionario}, com)\}\},$	
$\{\rho_{membro} \bowtie \rho_{presidente}\},$	% \mathcal{C}^{intra}
$\{\},$	% \mathcal{C}^{inter}
$\{\rho_{membro} \mapsto (3, 15), \rho_{presidente} \mapsto (1, 1), \rho_{secretario} \mapsto (1, 2),$	% np
$\rho_{candidato} \mapsto (0, 40)\},$	
$\{\}\rangle$	% ng

□

Exemplo 3.3 Para exemplificar o uso do modelo \mathcal{MOISE}^+ na definição de organizações hierarquizadas, uma outra alternativa de definição do grupo de seleção é apresentada na figura 3.4. Neste caso, comparando com o exemplo anterior, tem-se que (i) os funcionários não podem se comunicar; (ii) o secretário pode se comunicar somente com os membros da comissão de seleção; (iii) o candidato pode se comunicar somente com o secretário; e (iv) a comissão de seleção deve ser formada por exatamente cinco membros sendo um o presidente. O presidente passa a ter autoridade sobre os membros e sobre o secretário que tem autoridade sobre o candidato (!).

□

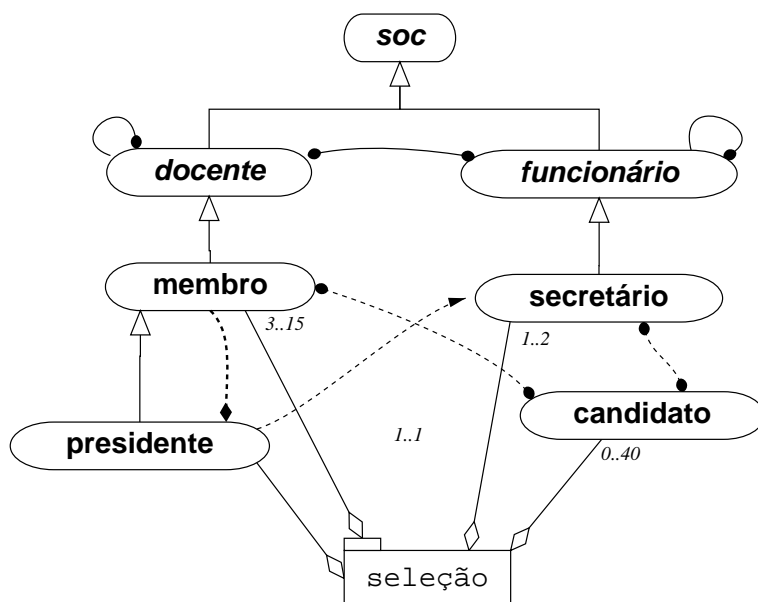


Figura 3.3: Exemplo de EE para seleção de candidatos.

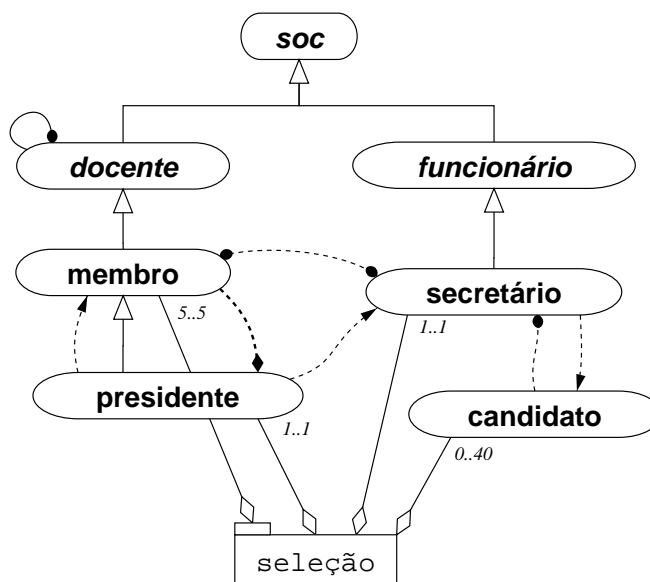


Figura 3.4: Outra possibilidade para EE de seleção de candidatos.

A partir destes exemplos⁶, pode-se perceber que algumas EE podem ser mais detalhadas, e eventualmente, mais restritivas que outras. No caso do grupo de seleção, certamente a segunda proposta (exemplo 3.3) é uma estrutura organizacional que restringe mais os comportamentos possíveis (um conjunto S menor — figura 3.1) dos agentes que a primeira proposta (exemplo 3.2). Por um lado, uma EE mais detalhada diminui o espaço de busca dos agentes pois o número de opções de ações possíveis é menor. Isso pode tanto tornar o mecanismo de decisão dos agentes mais rápido quanto garantir à sociedade um comportamento que leve à sua finalidade. Por outro lado, EE muito restritivas são difíceis de especificar, e não há como garantir que uma determinada EE irá levar o SMA para sua finalidade, perde-se a autonomia dos agentes e, conseqüentemente, a possibilidade de adaptação à pequenas mudanças no ambiente.

3.3 Especificação funcional

Se, por um lado, a EE procura limitar a autonomia dos agentes por meio de ligações entre papéis, não fica evidente como a finalidade da sociedade será alcançada a partir destas restrições. As restrições estruturais são vistas como contribuições na construção de um ambiente favorável à finalidade, mas que não determinam diretamente como alcançá-la. Por outro lado, um dos objetivos da organização é melhorar a eficiência da sociedade — o grau de desempenho com que a sua finalidade é alcançada. Uma maneira de melhorar a eficiência de uma sociedade é estabelecendo procedimentos para realizar certas atividades, assim como acontece nas sociedades observadas na natureza. Por exemplo, em uma escola há o procedimento de inscrição que se estabeleceu depois de ter sido testado, melhorado e validado. Este procedimento tornou-se bastante eficiente, tanto pelo fato de ter sido aperfeiçoado no decorrer do tempo quanto pelo fato de seus usuários terem se habituado a ele. Assim como este, muitos outros exemplos podem ser citados: procedimento de negociação, procedimento para obter um extrato bancário, procedimento de alteração de um jogador durante uma partida de futebol, procedimento de levar oxigênio às células, etc. Todos estes são procedimentos que amadureceram no decorrer da existência das sociedades,

⁶Outros exemplos serão apresentados nos capítulos 7 e 8. Em (HÜBNER; SICHMAN; BOISSIER, 2002a; HÜBNER; SICHMAN; BOISSIER, 2002b; HÜBNER; SICHMAN; BOISSIER, 2002c), o modelo MOISE⁺ é utilizado para especificar os grupos de um time de futebol.

tornando-se mais robustos e eficientes. Estes tipos de procedimento serão denominados de *esquemas sociais* (LUGO; HÜBNER; SICHMAN, 2001): um conjunto de metas estruturado por meio de planos. Na área de SMA esta dimensão funcional normalmente lida com os problemas de **coordenação** dos agentes por meio de noções como *Partial Global Plans* (DECKER; LESSER, 1992), *Shared Plans* (GROSZ; KRAUS, 1996), STEAMs (TAMBE, 1997) e *Joint Intentions* (COHEN; LEVESQUE, 1991).

No modelo MOISE^+ , a EF é constituída por um conjunto de esquemas sociais, além de uma relação de preferência entre missões. A seguir serão apresentados os conceitos necessários para a construção das noções de esquemas e missões.

3.3.1 Metas globais

A noção fundamental para os esquemas sociais, e conseqüentemente para a Especificação Funcional (EF), é a noção de *meta global*. Uma meta global representa um estado do mundo que é desejado pelo SMA. Uma meta global diferencia-se de uma meta local pelo fato desta última ser uma meta de um único agente, enquanto a primeira é uma meta de todo o SMA. Na terminologia proposta por Castelfranchi (1996), trata-se de um *objetivo social cooperativo*. No texto que segue, quando não é dito se a meta é global ou local, assume-se uma meta global.

A cada meta global g é associada uma combinação de três valores que indicam

1. seu *nível de satisfatibilidade*: indica se a meta já foi alcançada (valor *satisfied*) ou não (valor *unsatisfied*), ou ainda se ela é impossível de ser alcançada (valor *impossible*);
2. seu *nível de alocação*: indica se já existe ou não algum agente comprometido com a satisfação da meta (valores *committed* e *uncommitted*, respectivamente);
3. seu *nível de ativação*: indica se as pré-condições necessárias para que a meta seja satisfeita estão presentes (valores *permitted* e *forbiden*). Por exemplo, a meta “entregar a documentação” é *forbiden* até que a documentação esteja toda preparada.

O valor inicial de uma meta é (*unsatisfied, uncommitted, forbiden*) e seu valor vai

sendo alterado no decorrer do funcionamento do sistema. A dinâmica dos valores de uma meta será explicada nas seções seguintes e no capítulo 4.

Para simplificar a notação das fórmulas nas próximas seções, os seguintes predicado são definidos:

- $isSatisfied(\theta)$ é verdade se a meta θ estiver satisfeita;
- $isPossible(\theta)$ é verdade se a meta θ for possível de ser satisfeita.

3.3.2 Nível individual: missões

Um Esquema Social (ES) é constituído, no nível individual, por missões. 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. No exemplo da figura 3.5, o agente que se compromete com a missão m_4 deve tentar satisfazer as metas g_0 e g_{12} , já que o mesmo agente deve satisfazer estas duas metas.

Definição 3.11 (Missão) Sendo \mathcal{M} o conjunto de todas as missões e \mathcal{G}_{fs} o conjunto das metas globais da EF, uma missão $m \in \mathcal{M}$ é um sub-conjunto de metas globais

$$m \subset \mathcal{G}_{fs} \quad (3.29)$$

□

As metas podem ter argumentos que detalham melhor o seu estado de satisfação. Por exemplo, na meta g_1 , uma data (argumento Dt) deve ser instanciada detalhando o prazo final de entrega da documentação.

3.3.3 Nível coletivo: esquema social

Um ES é, essencialmente, uma árvore de decomposição de metas globais onde a raiz é a meta do ES.⁷ A decomposição de metas é feita por meio de planos

⁷A representação de ES proposta baseia-se nos seguintes trabalhos (HANNOUN et al., 2000; DECKER, 1996; DURFEE; LESSER, 1991; TAMBE, 1997; BREZILLON; PASQUIER; POMEROL, 2000). Como não se pretende uma especificação muito detalhada do funcionamento da organização, os esquemas, em geral, são simplificações das propostas feitas nestes trabalhos. Evidentemente, apesar de simples, os esquemas representam as informações que foram avaliadas como importantes na especificação de uma organização.

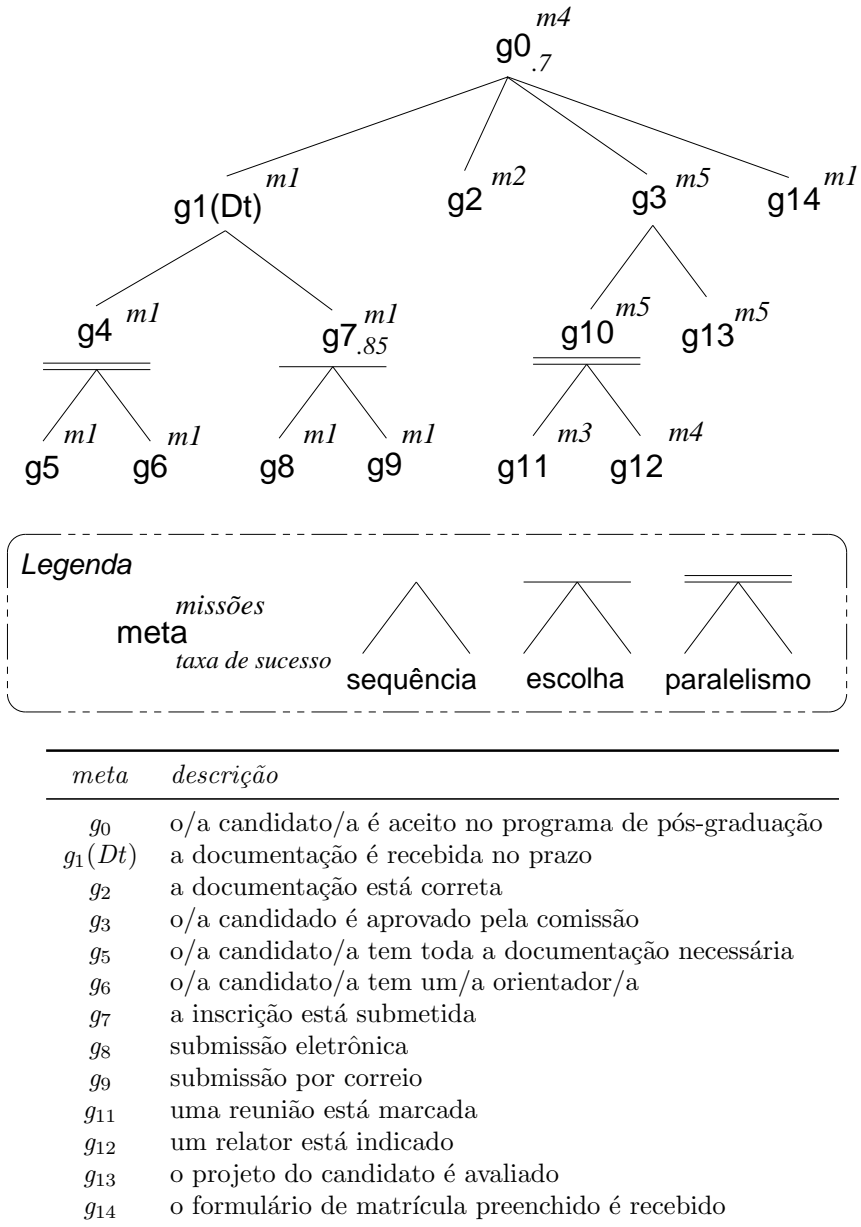


Figura 3.5: Exemplo de esquema social para ingresso em um curso de pós-graduação.

(denotados pelo operador $=$) que indicam uma forma de satisfazer uma meta. Por exemplo, no plano “ $g_0 = g_1, g_2, g_3, g_{14}$ ”, a meta g_0 é decomposta em quatro sub-metas indicando que ela será satisfeita como consequência do fato das sub-metas g_1, g_2, g_3, g_{14} também serem satisfeitas. No contexto de um plano específico, as metas que aparecem no lado direito do sinal $=$ deste plano são sub-metas da super-meta que aparece no lado esquerdo. Uma sub-meta em um plano pode ser super-meta em outro plano. Por exemplo, no plano “ $g_1 = g_4, g_7$ ”, g_1 é a super-meta e g_4 e g_7 são sub-metas deste plano. Portanto, o prefixo super/sub para uma meta é determinado no contexto de um plano.

Três tipos de operadores podem ser usados na construção de um plano:

- **seqüência** “,”: o plano “ $g_1 = g_4, g_7$ ” significa que a meta g_1 será satisfeita se a meta g_4 for satisfeita e depois a meta g_7 também o for. O nível de satisfatibilidade de g_1 é dado por⁸

$$isSatisfied(g_1) \Leftarrow isSatisfied(g_4) \wedge isSatisfied(g_7)$$

- **escolha** “|”: o plano “ $g_7 = g_8 | g_9$ ” significa que a meta g_7 será satisfeita se uma, e somente uma, das metas g_8 ou g_9 for satisfeita.

$$isSatisfied(g_7) \Leftarrow (isSatisfied(g_8) \wedge \neg isSatisfied(g_9)) \vee (isSatisfied(g_9) \wedge \neg isSatisfied(g_8))$$

- **paralelismo** “||”: o plano “ $g_4 = g_5 || g_6$ ” significa que a meta g_4 será satisfeita quando ambas as metas g_5 e g_6 o forem. Contudo, ao contrário da escolha, as duas sub-metas podem ser buscadas em paralelo.

$$isSatisfied(g_4) \Leftarrow isSatisfied(g_5) \wedge isSatisfied(g_6)$$

Conforme as necessidades da aplicação sendo especificada, a decomposição das metas pode ser descrita mais sucintamente através da omissão de metas intermediárias. No exemplo, tem-se que o plano para g_3 é escrito $g_3 = (g_{11} || g_{12}), g_{13}$, quanto formalmente tem-se dois planos: $g_3 = g_{10}, g_{13}$ e $g_{10} = g_{11} || g_{12}$.

⁸Não se está definindo aqui a semântica do operador de seqüência, apenas as implicações deste operador na satisfatibilidade de uma meta.

Para as metas que não têm planos em um ES, eventualmente pode existir um outro ES onde esta meta é a raiz. Caso não exista nem plano nem ES para uma meta, tal meta é chamada de *meta folha*. O agente que se comprometer com uma meta folha não tem na EF um plano que o auxilie na sua satisfação. Portanto, este agente tem que encontrar uma forma de satisfazer a meta por meios próprios.

O conjunto de vários planos forma a árvore do esquema (como no exemplo da figura 3.5) e determina algumas condições de coordenação entre as atividades dos agentes. Um agente não pode, portanto, buscar satisfazer as metas globais de suas missões em qualquer momento. No exemplo da figura 3.5, a meta g_2 somente pode ser permitida quanto a meta g_1 for satisfeita. Portanto, no contexto de um esquema, as metas de uma missão estão dentro de um plano que determina quando os agentes comprometidos com elas podem iniciar o processo de busca por satisfazê-las, isto é, quando a meta é *permitted*. Mais precisamente, se um agente α aceita a missão m_i , ele se compromete em satisfazer todas as metas desta missão ($g_j \in m_i$) e α buscará satisfazê-las somente quando a pré-condição de g_j estiver satisfeita. Esta pré-condição é inferida a partir do operador de seqüência (por exemplo, a meta g_{11} somente pode ser tentada depois que a meta g_2 já esteja satisfeita). Esta noção de pré-condições permite estabelecer alguns mecanismos de *coordenação* entre os agentes envolvidos na execução do esquema.

O fato de um agente assumir uma missão não impede que ele delegue suas metas (folhas ou não) para outros agentes. O que está estabelecido no ES é que o agente se compromete perante sua sociedade a ter a meta satisfeita, não importando como ele conseguirá isso. Por exemplo, nada impede que o candidato peça a outro agente para levar sua documentação ao correio, isto é, executar as ações que satisfazem a meta g_9 . Mas é responsabilidade do candidato fazer com que a meta seja satisfeita.

A noção de plano adotada no modelo \mathcal{MOISE}^+ não considera que caso as sub-metas sejam alcançadas, necessariamente a super-meta também o será. Por exemplo, a meta g_0 , com plano $g_0 = g_1, g_2, g_3, g_{14}$, provavelmente será satisfeita se as metas g_1, g_2, g_3 e g_{14} forem satisfeitas, mas não certamente. O ambiente interfere no sucesso de g_0 . Apesar da escola receber a documentação no prazo, a documentação estar correta, ter sido aprovada pela comissão de seleção e o formulário de matrícula ter sido entregue, o candidato/a pode não ser aceito

no programa de pós-graduação porque, por exemplo, o secretário esqueceu de encaminhar a matrícula para o departamento de administração acadêmica⁹. Por esta razão, a cada plano é associada uma probabilidade de sucesso, isto é, qual a crença da sociedade na possibilidade de sucesso do plano. Este valor é escrito como subscrito do operador = e, caso não seja informado na definição inicial do plano, é de 100%¹⁰. Por exemplo, o plano $g_7 =_{0.85} g_8 \mid g_9$ indica que se acredita que em 85% dos casos o plano para g_7 funciona.

Definição 3.12 (Plano) Formalmente, o conjunto de todos os planos é denotado por \mathcal{P} e um plano p é representado pela tupla

$$p = (\text{head}, \text{op}, \text{tx}, \mathcal{G}^n) \quad (3.30)$$

onde *head* é a super-meta do plano, *op* é o operador do plano ($\text{op} \in \{\text{,}, \mid, \parallel\}$), *tx* é a taxa de sucesso do plano e o último elemento é a seqüência de sub-metas. Para simplificar a leitura, o plano p é escrito na forma $\text{head} =_{\text{tx}} g_1 \text{ op } g_2 \text{ op } \dots g_n$ (g_i é o i -ésimo elemento da seqüência \mathcal{G}^n).

A função $\text{getHead} : \mathcal{P} \rightarrow \mathcal{G}$ projeta o primeiro elemento do plano, isto é, retorna a super-meta de um plano. \square

Definição 3.13 (Esquema Social) Formalmente, o conjunto de todos os esquemas sociais é denotado por \mathcal{SCH} e um esquema sch é representado pela tupla

$$\text{sch} = (\mathcal{G}, \mathcal{P}, \mathcal{M}, \text{mo}, \text{nm}) \quad (3.31)$$

onde:

- \mathcal{G} é o conjunto de metas do ES sch ¹¹;
- \mathcal{P} é o conjunto de planos que constrói a árvore de decomposição de metas do ES;
- \mathcal{M} é o conjunto de missões;

⁹Neste exemplo, poderia-se argumentar que o plano para g_0 não está completo e deveria incluir a meta “matrícula processada pelo departamento de administração acadêmica”. Contudo, nem sempre é possível (por falta de conhecimento do problema, por exemplo) definir os planos de forma suficientemente detalhada.

¹⁰No decorrer do funcionamento do SMA este valor é alterado pela razão entre o número de execuções com sucesso e o número total de execuções do plano

¹¹Assim como ocorre com o conjunto de papéis, denota-se $\text{sch}.g$ para indicar que se trata da meta g definida no esquema sch .

- $mo : \mathcal{M} \rightarrow \mathbb{P}(\mathcal{G})$ é uma função que determina o conjunto de metas de cada missão; e
- $nm : \mathcal{M} \mapsto \mathbb{N} \times \mathbb{N}$ determina o número (mínimo, máximo) de agentes que devem se comprometer em cada missão.

Como na EE, diz-se que um ES é bem formado se os valores da função nm são respeitados. Caso a função não tenha mapeamento para alguma missão, assume-se como cardinalidade o valor $(1, \infty)$, isto é, um ou infinitos agentes podem se comprometer com a missão.

Os agentes que se comprometem com a missão raiz do ES têm o direito de iniciar a execução do ES. \square

A partir do ES, pode-se definir formalmente uma meta folha como

$$isLeafGoal(g) \Leftarrow \nexists p \in \mathcal{P} \bullet getHead(p) = g \quad (3.32)$$

Exemplo 3.4 O ES da figura 3.5 teria a seguinte possível descrição:

$$\begin{aligned} sch_{ingresso} = & (\{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, g_{13}, g_{14}\}, \\ & \{“g_0 =_7 g_1, g_2, g_3, g_{14}” , \\ & “g_1 =_1 g_4, g_7” , \\ & “g_4 =_1 g_5 \parallel g_6” , \\ & “g_7 =_{.85} g_8 \mid g_9” \\ & “g_3 =_1 g_{10}, g_{13}” \\ & “g_{10} =_1 g_{11} \parallel g_{12}” \\ & \}, \\ & \{m_1, m_2, m_3, m_4, m_5\}, \\ & \{m_1 \mapsto \{g_1, g_4, g_5, g_6, g_7, g_8, g_9, g_{14}\}, \\ & m_2 \mapsto \{g_2\}, m_3 \mapsto \{g_{11}\}, m_4 \mapsto \{g_0, g_{12}\}, m_5 \mapsto \{g_3, g_{13}\}\}, \\ & \{m_1 \mapsto (0, 100), m_2 \mapsto (1, 1), m_3 \mapsto (1, 1), m_4 \mapsto (1, 1), \\ & m_5 \mapsto (3, 6)\}) \end{aligned}$$

Este ES é bem formado se, no máximo, 100 agentes se comprometerem com a missão $sch_{ingresso}.m_1$, se um, e somente um, agente se comprometer com as missões

$sch_{ingresso.m_2}$, $sch_{ingresso.m_3}$ e $sch_{ingresso.m_4}$, e se de três a seis agentes se comprometerem com a missão $sch_{ingresso.m_5}$ (aprovar o condidato). O agente que se comprometer com a missão $sch_{ingresso.m_4}$ pode criar uma instância deste ES.

Os agentes que se comprometem com a missão $sch_{ingresso.m_1}$ passam a ter, neste exemplo, várias metas (toda a sub-árvore abaixo de $sch_{ingresso.g_1}$ — ter a documentação recebida no prazo). Certamente se poderia optar por não ramificar a meta $sch_{ingresso.g_1}$ (tornando-a uma folha) e deixar que os agentes que vierem a se comprometer com ela desenvolvam um plano para satisfazê-la. A razão de decompor $sch_{ingresso.g_1}$, neste exemplo, é

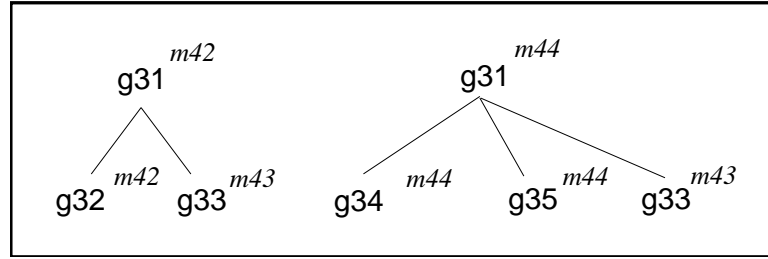
1. de determinar que o agente que se compromete com a missão m_1 **deve** satisfazer certas metas (ter toda a documentação, satisfazendo a meta g_5 , deve procurar um orientador g_6 , deve encaminhar a documentação g_7 , etc.), ele é responsável por elas e deve mostrar que as alcançou;
2. de determinar que o agente deve fazê-lo de um modo pré-determinado (nos planos); e
3. de ajudar o agente na obtenção de suas metas, já que um bom plano para sua realização já está estabelecido.

□

Exemplo 3.5 Um segundo exemplo bastante simples é o ES utilizado para realizar uma prova. Considerando as metas

<i>meta</i>	<i>descrição</i>
g_{31}	realizar uma prova
g_{32}	preparar a prova
g_{33}	responder a prova
g_{34}	encontrar uma prova aplicada nos anos anteriores
g_{35}	alterar o texto da prova

os seguintes ESs poderiam ser montados:



$$\begin{aligned}
 sch_{prova1} = & \langle \{g_{31}, g_{32}, g_{33}\}, \\
 & \{“g_{31} = g_{32}, g_{33}”\}, \\
 & \{m_{42}, m_{43}\}, \\
 & \{m_{42} \mapsto \{g_{31}, g_{32}\}, m_{43} \mapsto \{g_{33}\}\}, \\
 & \{m_{42} \mapsto (1, 1), m_{43} \mapsto (1, 100)\} \rangle
 \end{aligned}$$

$$\begin{aligned}
 sch_{prova2} = & \langle \{g_{31}, g_{34}, g_{35}, g_{33}\}, \\
 & \{“g_{31} = g_{34}, g_{35}, g_{33}”\}, \\
 & \{m_{44}, m_{43}\}, \\
 & \{m_{44} \mapsto \{g_{31}, g_{34}, g_{35}\}, m_{43} \mapsto \{g_{33}\}\}, \\
 & \{m_{44} \mapsto (1, 1), m_{43} \mapsto (1, 100)\} \rangle
 \end{aligned}$$

Foram definidos neste exemplo dois ES para a mesma meta de realizar uma prova ($sch_{prova1}.g_{31}$ e $sch_{prova2}.g_{31}$). \square

3.3.4 Preferência entre missões

Em alguns casos se deseja estabelecer uma *ordem de preferência* entre as missões. Por exemplo, para dizer que a missão m_{42} (preparar uma prova nova) é preferível à missão m_{44} (utilizar uma prova já aplicada). No modelo \mathcal{MOISE}^+ , esta ordem pode ser especificada da seguinte forma: $m_{42} \prec m_{44}$, definindo que a missão m_{42} é socialmente preferida. Se, em um determinado momento, um agente estiver comprometido com as missões m_{42} e m_{44} e existirem metas permitidas em ambas as missões, então ele deve dar preferências às metas da missão m_{42} .

Definição 3.14 (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 de todas as missões da EF, mesmo de missões de ES diferentes. As fórmulas seguintes são verdadeiras para a relação de preferência (anti-simetria e transitividade):

$$m_1 \prec m_2 \wedge m_2 \prec m_1 \Rightarrow m_2 = m_1 \quad (3.33)$$

$$m_1 \prec m_2 \wedge m_2 \prec m_3 \Rightarrow m_1 \prec m_3 \quad (3.34)$$

□

Nota-se que esta ordem de preferência entre missões pode, indiretamente, determinar uma ordem de preferência entre esquemas, como no exemplo 3.5, onde as duas missões são raízes de ES.

Retomando novamente a definição de especificação organizacional (EO) iniciada na página 38, o segundo elemento de uma EO, a especificação funcional, é definido a seguir.

Definição 3.15 (*Especificação Funcional*) Dadas as definições de ES e preferências, a EF de uma organização é um conjunto (\mathcal{S}) de ES e a relação de preferência entre as missões destes esquemas (\mathcal{PR}). Portanto, uma EF fs do conjunto de todas as EF, denotado por \mathcal{FS} , é dado pelo par

$$fs = (\mathcal{S}, \mathcal{PR}) \quad (3.35)$$

onde:

$$\mathcal{S} \subseteq \mathcal{SCH} \quad (3.36)$$

$$\mathcal{PR} \subseteq \mathcal{M}_{fs} \times \mathcal{M}_{fs} \quad (3.37)$$

O conjunto de todas as missões é dado por:

$$\mathcal{M}_{fs} = \bigcup_{sch \in \mathcal{S}} sch.\mathcal{M} \quad (3.38)$$

O conjunto de todas as metas é dado por:

$$\mathcal{G}_{fs} = \bigcup_{sch \in \mathcal{S}} sch.\mathcal{G} \quad (3.39)$$

□

Toda EF contém, por definição, o seguinte ES:

$$\begin{aligned} sch_{soc} = \langle & \{ \gamma, g_1, g_2, \dots, g_n \}, \\ & \{ \gamma = g_1 \parallel g_2 \parallel \dots \parallel g_n \}, \\ & \{ m_{soc} \}, \mathcal{GT}_s, \\ & \{ m_{soc} \mapsto \{ \gamma \} \}, \\ & \{ m_{soc} \mapsto (1, \infty) \} \rangle \end{aligned}$$

onde:

- γ representa a *finalidade* do SMA;
- g_i são as metas raízes dos ES de \mathcal{S} ($1 \leq i \leq n$);
- a finalidade γ é alcançada se todas as metas g_i forem alcançadas em paralelo, uma vez que todos os esquemas onde estas metas são raízes podem ser executados em paralelo;
- há somente uma missão m_{soc} com uma única meta: a finalidade do SMA; e
- pelo menos um agente deve se comprometer com esta missão.

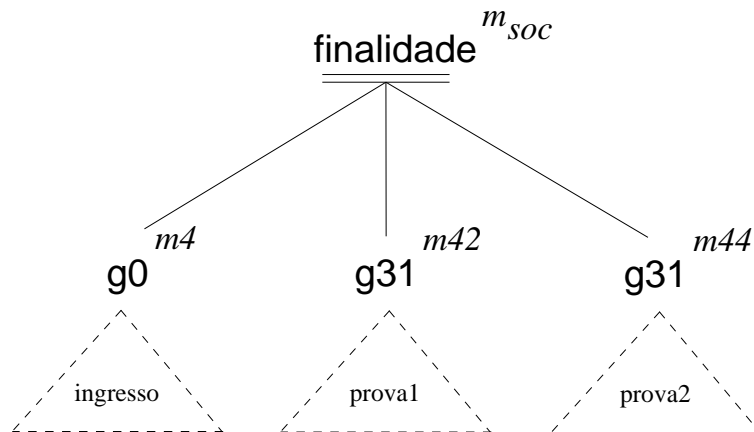
Este ES tem o objetivo de unificar todos os ES em uma única árvore de decomposição de metas, já que todas as folhas de sch_{soc} são raízes de outros ES.

Exemplo 3.6 A EF para a escola teria a seguinte descrição:

$$f_{escola} = (\{ sch_{ingresso}, sch_{prova1}, sch_{prova2}, sch_{soc} \}, \{ m_{42} \prec m_{44} \})$$

onde os três primeiros esquemas foram apresentados nos exemplos anteriores e o último ES é:

$$\begin{aligned} sch_{soc} = \langle & \{ \gamma, sch_{ingresso} \cdot g_0, sch_{prova1} \cdot g_{31}, sch_{prova2} \cdot g_{31} \}, \\ & \{ \gamma = sch_{ingresso} \cdot g_0 \parallel sch_{prova1} \cdot g_{31} \parallel sch_{prova2} \cdot g_{31} \}, \\ & \{ m_{soc} \}, \\ & \{ m_{soc} \mapsto \{ \gamma \} \}, \\ & \{ m_{soc} \mapsto (1, \infty) \} \rangle \end{aligned}$$



□

Resumidamente, uma EF descreve como um SMA normalmente alcança suas metas globais, isto é, como estas metas são decompostas por planos e distribuídas aos agentes por missões. Não é objetivo do modelo determinar como estes esquemas são definidos: eles podem tanto ser definidos pelo projetista do SMA que coloca nestes esquemas seu conhecimento do problema, como pelos próprios agentes que guardam na EF suas melhores soluções. Por exemplo, a primeira vez que um problema surge e não há ES para solucioná-lo, os agentes devem construir um plano. Se tal plano for bom, ele é guardado para a próxima vez que o mesmo problema surgir.

Nota-se que, para a descrição de uma EF, não é necessário fazer qualquer referência à EE (papéis e grupos, por exemplo). Esta independência é proposital e visa permitir que um SMA possa mudar sua EF sem alterar a EE e vice versa. Mesmo que, vistas individualmente, estas duas especificações possam ser concebidas/descritas/projetadas independentemente, no sistema como um todo essas duas partes devem ser ligadas. Por exemplo, é necessário dizer que tipo de agente pode (ou deve) se comprometer com uma missão. Esse é justamente o objetivo do terceiro elemento do modelo, mostrado a seguir.

3.4 Especificação deôntica

A relação entre a EE e a EF é feita pela Especificação Deôntica (ED) em seu nível individual, isto é, especifica-se quais as missões que um papel tem permissão ou

obrigação de se comprometer. Uma permissão $per(\rho, m, tc)$ determina que um agente com o papel ρ pode se comprometer com a missão m . Restrições temporais (tc) são estabelecidas para a permissão, isto é, se estabelece um conjunto de períodos de tempo onde a permissão é válida (por exemplo: todo dia a qualquer hora, somente nos domingos das 14h às 16h, no primeiro dia do mês, etc.)¹². Existe um conjunto, denotado por Any , que contém todos os períodos de tempo. Uma obrigação $obl(\rho, m, tc)$ estabelece que um agente com papel ρ é obrigado a se comprometer com a missão m nos períodos de tempos determinados em tc .

Para estes dois predicados as seguintes fórmulas são válidas:

$$obl(\rho, m, tc) \Rightarrow per(\rho, m, tc) \quad (3.40)$$

$$obl(\rho, m, tc) \wedge \rho \sqsubset \rho' \Rightarrow obl(\rho', m, tc) \quad (3.41)$$

$$per(\rho, m, tc) \wedge \rho \sqsubset \rho' \Rightarrow per(\rho', m, tc) \quad (3.42)$$

A primeira regra indica que quando um papel é obrigado a executar uma missão então ele também tem permissão para executá-la. As duas outras regras determinam o comportamento das relações deônticas quanto à herança.

Finalizando a definição de especificação organizacional (EO) iniciada na página 38, o terceiro elemento de uma EO, a especificação deôntica, é definido a seguir.

Definição 3.16 (Especificação Deôntica) A ED de uma organização é representada pela tupla

$$(\mathcal{P}, \mathcal{O}) \quad (3.43)$$

onde:

- \mathcal{P} é um conjunto de permissões

$$\mathcal{P} \subseteq \mathcal{R}_{ss} \times \mathcal{M}_{fs} \times \mathbb{P}(Any) \quad (3.44)$$

¹²A definição precisa da linguagem utilizada para expressar esse conjunto de períodos será aqui omitida por concisão, podendo ser consultada em Carron e Boissier (2001).

- \mathcal{O} é um conjunto de obrigações

$$\mathcal{O} \subseteq \mathcal{R}_{ss} \times \mathcal{M}_{fs} \times \mathbb{P}(Any) \quad (3.45)$$

Toda ED tem, pelo menos, a seguinte relação deôntica: $obl(\rho_{soc}, m_{soc}, Any) \in \mathcal{O}$. Como todos os papéis são especializações de ρ_{soc} , todos os agentes são obrigados a se comprometer com a missão m_{soc} (a missão raiz do ES sch_{soc}). Uma vez que sch_{soc} representa o plano global para alcançar a finalidade do SMA, esta obrigação implica que todos os agentes estejam comprometidos com a finalidade do sistema. \square

Exemplo 3.7 No exemplo da escola, parte da especificação deôntica poderia ser:

$$\begin{aligned} ds_{escola} = \langle & \{per(\rho_{presidente}, sch_{ingresso}.m_4, [feb/02\ feb/28]), \\ & per(\rho_{secretario}, sch_{ingresso}.m_2, Any), \\ & per(\rho_{secretario}, sch_{ingresso}.m_3, Any), \\ & per(\rho_{membro}, sch_{ingresso}.m_5, Any), \\ & per(\rho_{candidato}, sch_{ingresso}.m_1, Any) \\ & \}, \\ & \{obl(\rho_{aluno}, sch_{prova1}.m_{43}, Any) \\ & obl(\rho_{professor}, sch_{prova1}.m_{42}, periodic\ 3) \\ & \} \rangle \end{aligned}$$

Neste exemplo, o presidente da comissão pode iniciar, somente no mês de fevereiro, o ES $sch_{ingresso}$ porque ele tem permissão para a missão raiz deste ES (figura 3.5). Uma vez que o ES é criado, os outros agentes que assumiram os demais papéis da EE ($\rho_{membro}, \rho_{candidato}, \dots$) podem se comprometer com as missões do ES $sch_{ingresso}$. Ainda de acordo com a ED acima, se um agente assumir o papel de candidato, e existir uma instância de $sch_{ingresso}$, ele pode se comprometer com a missão m_1 (encaminhar sua submissão).

No caso do secretário e suas missões (m_2 e m_3), tem-se a seguinte situação: o grupo de seleção pode ter um ou dois secretários (conforme definido na EE do exemplo 3.2, página 52); e no esquema de ingresso cada uma das missões m_2 e m_3 pode ter no máximo um agente comprometido (EF do exemplo 3.4, página 62). Portanto, ou o mesmo secretário se compromete com as duas missões (o grupo teria necessidade de um único secretário), ou cada secretário se compromete com

somente uma das duas missões (o grupo teria dois secretários). Mas não é possível que haja dois secretários e cada um se comprometa com as duas missões. A EO não diz qual a melhor alternativa, os próprios agentes devem tomar essa decisão. Claro que, se fosse desejado, poderia-se facilmente forçar os agentes a adotar uma das duas situações acima. Bastaria, por exemplo, dizer que no grupo seleção pode haver somente um secretário; ou que no lugar das missões m_2 e m_3 , o secretário teria possibilidade de se comprometer somente a uma nova missão que tem cardinalidade $(1, 1)$ e as metas de m_2 e m_3 . O objetivo do modelo é permitir que várias situações possam ser especificadas.

As últimas duas obrigações

$$\begin{aligned} &obl(\rho_{aluno}, sch_{prova1}.m_{43}, Any) \text{ e} \\ &obl(\rho_{professor}, sch_{prova1}.m_{42}, periodic\ 3) \end{aligned}$$

determinam, respectivamente, que os alunos são obrigados a se comprometer com a missão de realizar a prova (m_{43}) no tempo que a prova durar e que os professores devem realizar três provas (missão m_{42}) no período em que durar uma turma. Na entidade organizacional criada com esta especificação deôntica, esta última relação significa que um agente α que assumiu o papel de professor na turma $gr1$ (criada a partir da especificação gt_{turma}) é obrigado a fazer três provas durante a existência da turma $gr1$. \square

No contexto da figura 3.1 (página 38), a ED estabelece o conjunto $S \cap F$. Entre os comportamentos possíveis (S), um agente pode optar por um comportamento em $S \cap F$ porque (i) é obrigado a fazê-lo em função de uma obrigação dos papéis que assumiu ou (ii) porque participar de um esquema lhe dá um certo tipo de poder. Se um agente inicia um ES (isto é, um comportamento em $S \cap F$) ele pode forçar os demais, pela ED, a se comprometer com as outras missões do ES. Por exemplo, em um grupo turma, onde agentes assumiram os papéis de professor e aluno, se o agente professor se comprometer com a missão de realizar uma prova, os alunos são forçados pela ED a também se comprometerem com o esquema (realizando a prova). Portanto, o professor tem o poder de obrigar os alunos a participar da prova por meio do esquema sch_{prova1} . Deve-se notar que a ED determina as permissões e obrigações para com as metas globais estabelecidas. Quanto às metas locais, ou mesmo metas globais não consideradas na

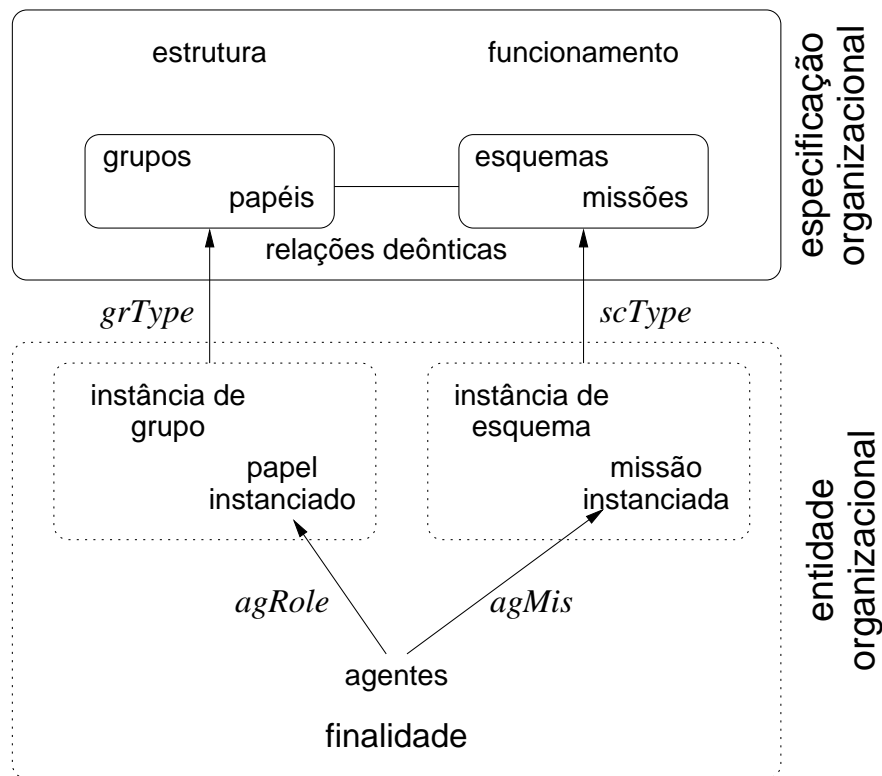


Figura 3.6: Visão simplificada dos constituintes de um SMA segundo o modelo \mathcal{MOISE}^+ .

especificação, a ED não faz qualquer restrição, isto é, os agentes têm completa autonomia para com as metas não globais.

3.5 Entidade organizacional

Uma especificação organizacional, formada pelas EE, EF e ED, não inclui agentes pois tem um caráter mais abstrato onde os agentes são representados por papéis. Em um SMA concreto há, portanto, a necessidade de que esta especificação seja instanciada por um conjunto de agentes formando uma Entidade Organizacional, onde fica estabelecida a posição destes agentes no contexto de uma EO (figura 3.6).

Definição 3.17 (Entidade Organizacional) Uma EnO é formada por uma finalidade que deve ser mantida por um conjunto de agentes que instanciam uma

EO. O estado de uma EnO em um instante t é denotado pela tupla

$$oe = (\gamma, os, \mathcal{A}, \mathcal{GI}, grType, subGr, agRole, \mathcal{SI}, scType, agMis, gState)^t \quad (3.46)$$

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;
- $grType : \mathcal{GI} \rightarrow \mathcal{GT}$ determina as especificações dos grupos em \mathcal{GI} ;
- $subGr : \mathcal{GI} \rightarrow \mathbb{P}(\mathcal{GI})$ determina os sub-grupos dos grupos;
- $agRole : \mathcal{A} \mapsto \mathbb{P}(\mathcal{R}_{ss} \times \mathcal{GI})$ 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;
- $scType : \mathcal{SI} \rightarrow \mathcal{S}_{fs} \times \mathbb{P}(\mathcal{GI})$ determina a especificação dos ES criados e também quais os grupos que criaram o ES; e
- $agMis : \mathcal{A} \mapsto \mathbb{P}(\mathcal{M}_{fs} \times \mathcal{SI})$ representa o conjunto de missões com que cada agente está comprometido.
- $gState : \mathcal{SI} \times \mathcal{G}_{fs} \mapsto \mathcal{GV}$ representa o estado das metas dos esquemas, ou seja, um valor do conjunto \mathcal{GV} dado por

$$\begin{aligned} \mathcal{GV} = & \{ \{unsatisfied, satisfied, impossible\} \times \\ & \{uncommitted, committed\} \times \\ & \{forbiden, permitted\} \} \end{aligned}$$

□

Exemplo 3.8 Considerando o exemplo que vem sendo construído para uma escola, uma instância desse tipo de sociedade em um determinado instante inicial 0 poderia ser:

```
oe0escola =  
( % finalidade  
  'formar pesquisadores e professores',  
  
  % especificação  
  (ssescola, fsescola, dsescola),  
  
  % agentes  
  {Antonio, Jaime, Anna, Jomi, Gustavo,  
   Reinaldo, Carlota, Fabricio, Marcia, Marcos},  
  
  % grupos  
  {},  
  
  % grType  
  {},  
  
  % subGg  
  {},  
  
  % agRole  
  {},  
  
  % esquemas  
  {},  
  
  % schType  
  {},  
  
  % agMis  
  {},  
  
  % gState  
  {})
```

Tem-se um conjunto de dez agentes que criaram uma EnO para atingir um objetivo comum ('formar pesquisadores e professores'), mas no início ainda não criaram grupos e, portanto, não assumiram papéis. A EnO criada possui a seguinte organização:

- a estrutura desta sociedade é ss_{escola} , definida na página 48;
- o funcionamento é dado por fs_{escola} , definido na página 66; e
- as relações deônticas são dadas por ds_{escola} , definido na página 69.

□

Exemplo 3.9 Considerando ainda a mesma escola, em outro momento t ,¹³ a descrição de sua EnO poderia ser¹⁴:

$$oe_{escola}^t =$$

$$\left(\begin{array}{l} \% \text{ finalidade} \\ \text{'formar pesquisadores e professores'}, \\ \\ \% \text{ especificação} \\ (ss_{escola}, fs_{escola}, ds_{escola}), \\ \\ \% \text{ agentes} \\ \{Antonio, Jaime, Anna, Jomi, Gustavo, \\ \quad Reinaldo, Carlota, Fabricio, Marcia, Marcos\}, \\ \\ \% \text{ grupos} \\ \{gr1_{escola}, gr1_{turma}, gr2_{turma}, gr3_{turma}, \\ \quad gr1_{corpodoc}, gr1_{selecao}\}, \\ \\ \% \text{ grType} \\ \{gr1_{escola} \mapsto gt_{escola}, \end{array} \right.$$

¹³A EnO tem uma história formada pelas mudanças no seu estado. O capítulo 4 irá detalhar como tais mudanças ocorrem.

¹⁴Para facilitar a leitura, será utilizada a seguinte convenção: $gr1_{escola}$ indica um grupo (gr) criado a partir da especificação de grupo escola (subscrito) e esta é a instância 1 de escola.

$$\begin{aligned} gr1_{turma} &\mapsto gt_{turma}, gr2_{turma} \mapsto gt_{turma}, gr3_{turma} \mapsto gt_{turma}, \\ gr1_{corpodoc} &\mapsto gt_{corpodoc}, gr1_{selecao} \mapsto gt_{selecao} \}, \end{aligned}$$

% subGg

$$\{gr1_{escola} \mapsto \{gr1_{turma}, gr2_{turma}, gr3_{turma} \mapsto gt_{turma}, \\ gr1_{corpodoc}, gr1_{selecao} \}\},$$

% agRole

$$\begin{aligned} Antonio &\mapsto \{(\rho_{diretor}, gr1_{corpodoc}), (\rho_{membro}, gr1_{selecao})\}, \\ Jaime &\mapsto \{(\rho_{professor}, gr1_{corpodoc}), (\rho_{membro}, gr1_{selecao}), \\ &\quad (\rho_{professor}, gr1_{turma})\}, \\ Anna &\mapsto \{(\rho_{professor}, gr1_{corpodoc}), (\rho_{presidente}, gr1_{selecao}), \\ &\quad (\rho_{professor}, gr2_{turma})\}, \\ Jomi &\mapsto \{(\rho_{aluno}, gr1_{turma}), (\rho_{aluno}, gr3_{turma})\}, \\ Gustavo &\mapsto \{(\rho_{aluno}, gr1_{turma}), (\rho_{aluno}, gr2_{turma})\}, \\ Reinaldo &\mapsto \{(\rho_{aluno}, gr2_{turma}), (\rho_{aluno}, gr3_{turma})\}, \\ Carlota &\mapsto \{(\rho_{aluno}, gr2_{turma}), (\rho_{professor}, gr3_{turma})\}, \\ Fabricio &\mapsto \{(\rho_{candidato}, gr1_{selecao})\}, \\ Marcia &\mapsto \{(\rho_{secretario}, gr1_{selecao})\} \}, \end{aligned}$$

% esquemas

$$\{si1_{prova1}, si2_{prova1}\},$$

% schType

$$\{si1_{prova1} \mapsto (sch_{prova1}, \{gr1_{turma}\}), si2_{prova1} \mapsto (sch_{prova1}, \{gr2_{turma}\})\},$$

% agMis

$$\begin{aligned} Jaime &\mapsto \{(sch_{prova1}.m42, si1_{prova1})\}, \\ Jomi &\mapsto \{(sch_{prova1}.m43, si1_{prova1})\}, \\ Gustavo &\mapsto \{(sch_{prova1}.m43, si1_{prova1})\}, \\ Anna &\mapsto \{(sch_{prova1}.m42, si2_{prova1})\}, \\ Gustavo &\mapsto \{(sch_{prova1}.m43, si2_{prova1})\}, \\ Carlota &\mapsto \{(sch_{prova1}.m43, si2_{prova1})\} \\ &\} \end{aligned}$$

```

% gState
{(si1prova1, g31) ↦ (unsatisfied, committed, permitted),
 (si1prova1, g32) ↦ (satisfied, committed, permitted),
 (si1prova1, g33) ↦ (unsatisfied, committed, permitted),
 (si2prova1, g31) ↦ (unsatisfied, committed, permitted),
 (si2prova1, g32) ↦ (unsatisfied, committed, permitted),
 (si2prova1, g33) ↦ (unsatisfied, uncommitted, forbidden)}
)

```

sendo que nesta entidade

- há um grupo escola com cinco sub-grupos (três turmas, um corpo docente e uma comissão de seleção);
- a partir de *agRole*, pode-se dizer que
 - uma turma tem por professor o agente Jaime e por alunos os agentes Jomi e Gustavo. Dados estes papéis e as ligações de *ss_{escola}* (veja a figura 3.2, página 49), o agente Jaime tem autoridade sobre os agentes Jomi e Gustavo nesse grupo, e os agentes Jomi e Gustavo podem se comunicar com qualquer aluno e com o agente Jaime (professor na turma onde são alunos). Os alunos (Jomi, Gustavo, Reinaldo e Carlota) podem conhecer Antonio (o diretor) mas não conversar com ele;
 - outra turma tem por professor o agente Anna e por alunos os agentes Gustavo, Reinaldo e Carlota;
 - a terceira turma *gr3_{turma}* tem como professor o agente Carlota e como alunos os agentes Jomi e Reinaldo. Carlota pode ser tanto professor em *gr3_{turma}* quanto aluno em *gr2_{turma}* porque há uma compatibilidade inter-grupo entre estes dois papéis;
 - o corpo docente é formado pelos agentes Antonio, Jaime e Anna, sendo Antonio o diretor. Carlota — professora da turma *gr3_{turma}* — não faz parte do corpo docente, isto é, assumir o papel de professor em uma turma não implica em assumir o mesmo papel em outros grupos (observa-se que o agente Jaime, para ser professor nos dois grupos, tem duas adoções de papel na função *agRole*);

- o grupo de seleção tem como presidente o agente Anna, como membros os agentes Antonio e Jaime, o agente Marcia como secretária e agente Fabricio como candidato;
- o agente Marcos não está em nenhum grupo;
- os grupos do tipo turma não estão bem formados já que possuem menos de cinco alunos. Já o grupo escola está bem formado (possui um sub-grupo corpo docente e menos de 300 alunos), assim como o grupo corpo docente. O grupo de seleção não está bem formado pois não possui três membros.

Nota-se que a adoção de papéis atende os requisitos de compatibilidade de papéis e que não há qualquer agente com papel abstrato;

- dois ES estão em execução no tempo t : duas instâncias do sch_{prova1} (definido na página 63), ambos tendo como grupo responsável $gr1_{turma}$ e $gr2_{turma}$, respectivamente;
- a partir de $agMis$ pode-se dizer que
 - nos dois esquemas, os professores assumem a missão m_{42} (preparar a prova) e os alunos a missão m_{43} (responder à prova). No caso do ES $si1_{prova1}$, o agente Jaime é o professor (considerado o grupo $gr1_{turma}$) e os agentes Jomi e Gustavo são alunos (considerado o mesmo grupo);
 - o agente Reinaldo, que é aluno no grupo $gr2_{turma}$, não está realizando a prova. Entretanto este agente deveria realizar a prova pois na ds_{escola} foi especificado que $obl(\rho_{aluno}, m_{43}, Any)$ e Reinaldo é aluno, logo, este agente não está cumprindo com uma de suas obrigações.

□

3.6 Comparação com outros modelos organizacionais

Terminada a definição do modelo \mathcal{MOISE}^+ , convém compará-lo com outros modelos organizacionais semelhantes, principalmente aqueles que influenciaram de

forma mais direta o seu desenvolvimento. Comparando-o com o MOISE, observa-se inicialmente que os seguintes componentes do MOISE não foram incluídos no MOISE^+ (a tabela 3.1 resume esta comparação):

1. as ações nos planos;
2. os recursos das missões; e
3. e a identificação do protocolo nas ligações de comunicação.

A razão da exclusão destes componentes é primeiramente de simplificar o modelo permitindo (i) focalizar somente alguns componentes organizacionais (estrutura, funcionamento e obrigações) e (ii) manter o modelo centralizado na organização (já que as ações, por exemplo, dizem respeito à especificação dos agentes e não da organização).

Por outro lado, o modelo MOISE^+ incluiu novos atributos ao modelo:

1. foi adicionada a relação de herança entre papéis para simplificar a descrição da organização;
2. foi adicionada a relação de compatibilidade entre papéis para limitar os papéis que um agente pode assumir numa sociedade ou grupo,
3. foi adicionada a noção de cardinalidade e conseqüentemente o atributo de “bem formado” aos grupos e esquemas;
4. as ligações e a compatibilidade passam a ter um escopo (intra ou inter-grupo);
5. os grupos podem ter sub-grupos;
6. adicionou-se a noção de esquema social que permite ver os planos em um contexto global e não no contexto de um papel como no modelo MOISE. Da mesma forma, a coordenação entre os agentes é vista globalmente e não localmente. Em outras palavras, as missões e os planos passam a pertencer a um ES e não a um papel;
7. a relação entre papel e missão passa a ser definida pelas relações deônticas (na verdade o MOISE também tem estas relações, contudo o pode/deve do

MOISE aplica-se às metas, ações e recursos; no \mathcal{MOISE}^+ aplica-se somente a metas);

8. as relações deônticas passam a ter um tempo de validade;
9. a ordem de preferência entre as missões pode ser determinada; e
10. a EnO passa a ter uma finalidade.

Em uma comparação do modelo \mathcal{MOISE}^+ com o modelo TÆMS (PRASAD et al., 1996), não se pode apresentar conclusões muito detalhadas, já que o foco do TÆMS está nas tarefas e o do \mathcal{MOISE}^+ está nos papéis. Portanto, enquanto o \mathcal{MOISE}^+ permite um detalhamento dos papéis sociais e suas relações melhor que o TÆMS (as noções da grupo, ligações entre papéis e compatibilidade do \mathcal{MOISE}^+ não existem no TÆMS), o TÆMS permite descrever os ES (tarefas) mais detalhadamente que o \mathcal{MOISE}^+ . As interferências entre tarefas (facilita, inibe, dispara, por exemplo) e algumas formas de relação entre tarefa e sub-tarefas do TÆMS não encontram similar no \mathcal{MOISE}^+ . Estes componentes tem como principal função auxiliar os algoritmos de escalonamento. Uma vez que o \mathcal{MOISE}^+ não tem esse objetivo, a falta destes componentes não compromete o modelo proposto no seu propósito de auxiliar os processos de reorganização.

3.7 Conclusões

O modelo organizacional proposto apresenta uma concepção de como a organização contribui para a finalidade do SMA, isto é, através de uma estrutura de ligações entre papéis e um conjunto de planos globais (os esquemas). Esta proposta apresenta as seguintes características principais que delimitam seu domínio de aplicação:

- i)* É uma concepção centrada na organização (conforme definido no capítulo 2). Não se pretende especificar os agentes e nem estabelecer qualquer requisito para eles. Esta é uma necessidade em SMA abertos, onde não se tem controle sobre o projeto dos agentes, eventualmente desenvolvidos para várias finalidades, por vários projetistas ou mesmo em várias linguagens de programação distintas. Logo, não se tem como saber (ou impor) a arquitetura interna destes agentes.

Tabela 3.1: Comparação entre os modelos MOISE e o $\mathcal{M}OISE^+$

<i>Componente</i>	MOISE	$\mathcal{M}OISE^+$
papel	definido como um conjunto de missões	é um tipo primitivo sobre o qual são estabelecidas restrições estruturais (equivalentes às ligações do MOISE) e funcionais (equivalentes às missões do MOISE)
missão	é uma obrigação ou permissão para um conjunto de metas, planos, ações e recursos	é parte de um esquema social e, como tal, é uma estrutura de decomposição de metas através de planos (não foram consideradas as ações e os recursos). A relação de obrigação e permissão também existe, não é definida na própria missão, mas na sua relação com a estrutura social
ligações grupo	são restrições sobre os papéis conjunto de papéis, missões e ligações	idem conjunto de papéis, ligações e sub-grupos com as cardinalidades. A especificação das missões não faz mais parte de um grupo.
instanciação	é um conjunto de pares (agente, papel), um conjunto de grupos instanciados e um conjunto de ligações instanciadas	idem, exceto a inclusão da finalidade e a adequação às mudanças no modelo

- ii)* Permite descrever uma organização em três dimensões: estrutural, funcional e deontica (conforme ilustra a figura 3.7). Em cada dimensão é possível definir um escopo individual (como papéis e missões) e um escopo coletivo (como grupos e esquemas).
- iii)* A noção de papel do modelo \mathcal{MOISE}^+ permite conceber um papel de forma semelhante à proposta de Castelfranchi (1996) onde as seguintes características são importantes:
 - a) papel é coletivo: um papel é assumido e existe em relação a um grupo. O papel não está no agente, mas na sua relação com um grupo (não é, por exemplo, um conjunto de capacidades internas do agente). No \mathcal{MOISE}^+ isso fica evidente quando o papel e suas relações são definidos dentro de um grupo (ver definição de grupo, página 45);
 - b) um papel é normativo: um papel é um conjunto de obrigações estabelecidas nas ligações entre papéis e nas suas relações deonticas (um papel não é uma alocação de tarefas).
- iv)* Permite estabelecer restrições sobre a dinâmica de formação da EnO através da noção de “bem formado” dos grupos e missões.
- v)* Torna explícita a finalidade do sistema.
- vi)* Permite a especificação dos aspectos organizacionais (papéis, planos, ...) de forma *independente* (devidamente ligados posteriormente por relações deonticas). Esta independência permite a mudança de um dos aspectos sem grandes conseqüências nos outros aspectos. Esta característica é particularmente interessante quando se considera o processo de reorganização.

Mesmo que os exemplos apresentados no desenvolvimento do \mathcal{MOISE}^+ eventualmente sugeriram que os agentes da EnO são cognitivos, isso não é uma necessidade. O foco do modelo está na organização da sociedade e não nos agentes que a constituem, que podem ter qualquer tipo de arquitetura interna de funcionamento. Portanto, é possível descrever a organização de sociedades formadas por agentes reativos (como formigas, por exemplo). Nestes casos, o modelo serve para que um observador (humano ou computacional) da sociedade possa fazer

uma descrição da organização e não para que um projetista (humano ou computacional) estabeleça a organização que deverá ser respeitada pelos agentes. No contexto apresentado na figura 2.2 (página 16), o modelo \mathcal{MOISE}^+ pode ser utilizado tanto para descrever a organização institucionalizada (tipo OR e OC) quanto a observada (tipo AR e AC). Em outras palavras, apesar de não ser o objetivo deste trabalho, o \mathcal{MOISE}^+ pode ser utilizado como uma linguagem organizacional formal.

Apesar desta flexibilidade de utilização do modelo proposto, nesta tese ele será utilizado por um conjunto de agentes que deseja alterar sua organização corrente (descrita no \mathcal{MOISE}^+), tratando-se, portanto, de uma sociedade do Tipo OC da figura 2.2 (os agentes têm capacidade de representar sua organização). Certamente, as propriedades de uma descrição organizacional, e do modelo organizacional subjacente à ela, interfere na eficiência do processo de reorganização pois é sobre tal descrição, e o que ela pode ou não representar, que os agentes irão raciocinar no momento de avaliar a organização corrente e de projetar uma nova organização. Neste sentido, o modelo proposto é importante pois (i) permite a especificação de vários tipos de organização (como visto nos exemplos e ainda será visto nos estudos de caso) e (ii) tal especificação permite que os agentes alterem apenas uma das dimensões da organização.

Contudo, antes de detalhar o processo de reorganização (assunto da parte II), é necessário compreender melhor como um SMA organizado segundo o modelo \mathcal{MOISE}^+ pode ser implementado e funcionar. O capítulo seguinte irá justamente apresentar uma arquitetura para um SMA que funcione segundo este modelo.

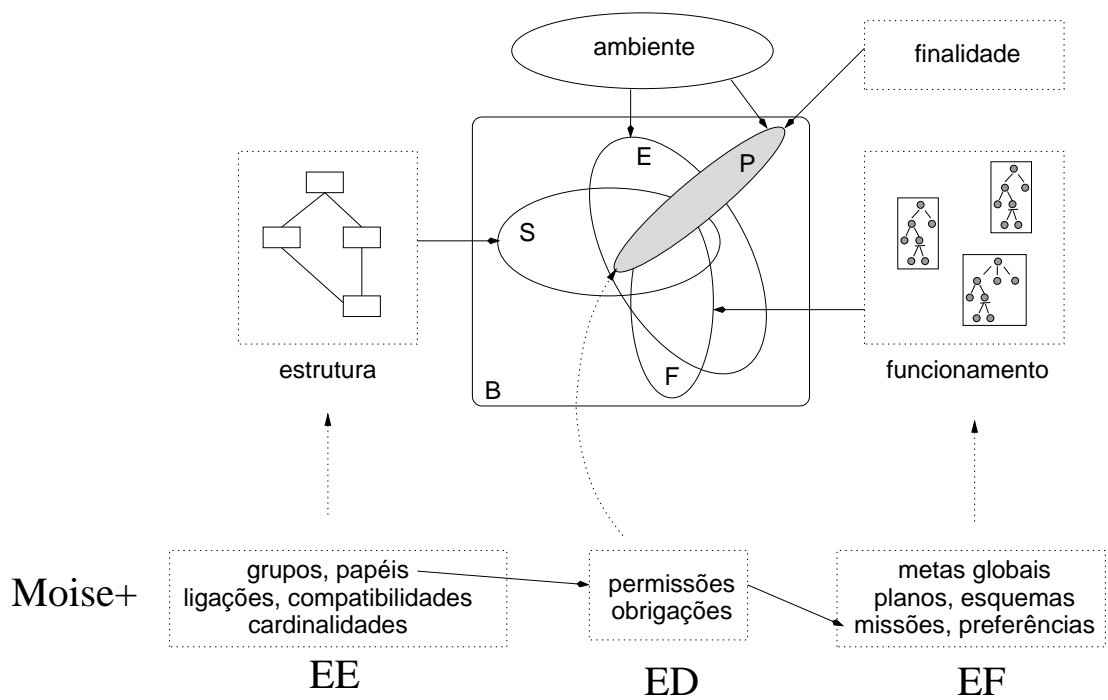


Figura 3.7: Resumo do modelo proposto segundo a visão organizacional adotada.

4 Proposta de arquitetura para o modelo \mathcal{MOISE}^+

Enquanto o capítulo anterior descreveu *o que é* o modelo organizacional proposto, este capítulo apresenta *como funciona* um SMA que utiliza o modelo \mathcal{MOISE}^+ . Questões como as seguintes são respondidas no decorrer deste capítulo:

- Como funciona uma sociedade que segue o \mathcal{MOISE}^+ ?
- Com que linguagem se descreve a organização do SMA?
- Onde as informações organizacionais (o estado da EnO) ficam armazenadas (em um único lugar ou descentralizado)?
- Como o modelo interfere no comportamento dos agentes?
- Como se dá a coordenação dos agentes envolvidos na execução de um esquema?

Convém ressaltar que é proposta *uma* arquitetura para o modelo proposto e que portanto outras seriam possíveis. A arquitetura organizacional aqui proposta foca em organizações do tipo OC (figura 2.2), onde há uma organização institucionalizada com uma representação explícita e os agentes têm capacidade de raciocinar sobre tal representação. Não é objetivo desta arquitetura, por exemplo, oferecer suporte para agentes que queiram obter uma descrição da organização observada (organizações do tipo AC). Apesar do modelo \mathcal{MOISE}^+ permitir exprimir uma organização observada, esta arquitetura não busca suportar o processo de observação.

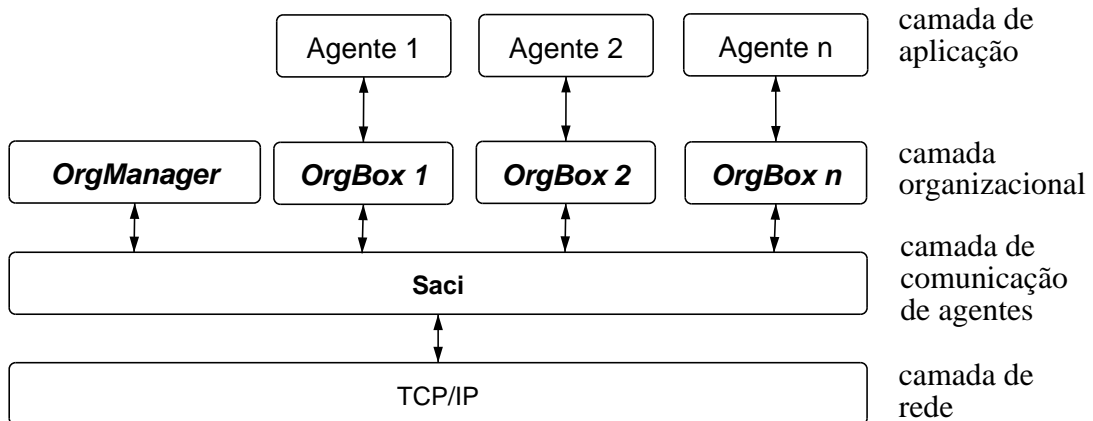


Figura 4.1: Camadas e componentes da arquitetura proposta.

4.1 Visão geral

A arquitetura proposta baseia-se em uma abordagem de camadas. Esta abordagem tem como principal vantagem a independência entre as camadas, onde cada camada obtém serviços da camada inferior e provê serviços para a camada superior. São propostas as seguintes camadas (figura 4.1):

Camada de rede: viabiliza a conexão entre os computadores onde os agentes executam.

Camada de comunicação entre agentes: viabiliza a comunicação entre os agentes por meio de uma Linguagem de Comunicação entre Agentes (LCA). A LCA utilizada será o KQML (FININ et al., 1994) implementada pelo ambiente SACI (HÜBNER; SICHMAN, 2000). O SACI também dá suporte ao controle da execução distribuída de agentes. No anexo A, é apresentado um breve resumo desta ferramenta.

Camada organizacional: oferece os serviços de manutenção do estado da EnO e o controle do cumprimento das regras estabelecidas pela organização. Esta camada, que será detalhada neste capítulo, é formada por dois componentes: *OrgManager* e *OrgBox*.

O *OrgManager* é um agente cuja função é manter o estado da EnO consistente, por exemplo, não deixando um agente assumir dois papéis incompatíveis. Toda mudança na EnO (entrada de um agente, criação de um

grupo, adoção de um papel, etc.) deve passar por este agente.¹ A seção 4.2 detalha o funcionamento deste agente.

O OrgBox é uma interface que os agentes utilizam para acessar a organização e os demais agentes. Sempre que um agente deseja realizar uma ação sobre a EnO (se comprometer com uma missão, por exemplo) ou enviar uma mensagem, ele deve solicitar este serviço ao seu OrgBox que, por sua vez, irá interagir, via SACI, com o OrgBox do outro agente. A seção 4.3 detalha o funcionamento deste componente.

Camada de aplicação: nesta camada estão os agentes da aplicação sendo desenvolvida. Como cada domínio de aplicação demanda diferentes soluções neste nível, este capítulo não dará muita atenção aos problemas desta camada, procurando definir a camada organizacional de forma independente da aplicação. Nos estudos de caso (parte III), este assunto será melhor explorado.

4.2 Dinâmica da entidade organizacional

A dinâmica de uma EnO consiste das alterações no estado da entidade causados por *eventos organizacionais*. Tais eventos ocorrem quando o OrgManager recebe uma mensagem de um agente solicitando a alteração de algum elemento da EnO. Somente a alteração de alguns dos elementos da EnO serão considerados, alterações na finalidade γ ou na especificação organizacional da entidade não serão consideradas nesta seção. Na verdade, alterações nestes dois elementos também são eventos organizacionais, contudo, eles serão considerados aqui como um caso particular e desenvolvidos no capítulo 6 dentro de um contexto de reorganização.

A EnO passa, portanto, a ter uma história representada por uma seqüência de estados. EnO^t denota o t -ésimo elemento desta seqüência que, se alterado por um evento organizacional, se transforma no elemento EnO^{t+1} . Caso não haja a marcação de tempo t em um elemento, assume-se o último estado da sua história, isto é, o estado corrente. Por exemplo, \mathcal{A} (o conjunto de agentes da EnO) é uma abreviação de \mathcal{A}^t .

¹Este agente é inspirado no agente *Facilitator* do KQML, que mantém o estado das páginas brancas (nome dos agentes) e amarelas (serviços dos agentes) do sistema.

Para simplificar a definição das condições para a ocorrência dos eventos organizacionais, as seguintes funções são definidas (estas funções podem ser facilmente definidas a partir das funções definidas no capítulo 3):

- $grPlayers(gr, \rho)$: retorna o conjunto dos agentes com papel ρ no grupo gr ,
- $schPlayers(si, m)$: retorna o conjunto dos agentes com a missão m na instância de esquema si ,
- $subgroups(gr, gt)$: retorna o conjunto dos sub-grupos do tipo gt que existem no grupo gr ,
- $superGr(gr)$ retorna a identificação do grupo onde gr é sub-grupo,
- $getRootGoal(si)$ retorna a meta raiz do esquema si ,
- $first(P)$ sendo P um par do tipo (x, y) , retorna x , e
- $second(P)$ sendo P um par do tipo (x, y) , retorna y .

A seguir é feita uma descrição detalhada dos eventos organizacionais com o objetivo de descrever como estes eventos alteram o estado da EnO mantido pelo OrgManager.

4.2.1 Criação da entidade organizacional

A criação da EnO consiste da criação do agente OrgManager e sua inclusão no sistema. A sua criação necessita de dois argumentos: a finalidade do sistema (parâmetro γ da EnO) e a EO². Segue o detalhamento deste evento.

²Na implementação realizada, a EO é descrita em um arquivo XML como exemplifica o anexo B.

Evento	criação de uma EnO
Argumentos	γ : finalidade da EnO os : especificação da EnO
Pré-condições	
Efeitos	$EnO^1 = \langle \gamma, os, \{OrgManager\}, \{\}, \{\}, \{\}, \{\}, \{\}, \{\}, \{\}, \{\} \rangle$ isto é, $\mathcal{A}^1 = \{OrgManager\}$ $\mathcal{GI}^1 = \{\}$ $grType^1 = \{\}$ $subGr^1 = \{\}$ $agRole^1 = \{\}$ $\mathcal{SI}^1 = \{\}$ $scType^1 = \{\}$ $agMis^1 = \{\}$ $gState^1 = \{\}$

Como consequência da criação de uma EnO, uma nova EnO é criada onde apenas os três primeiros argumentos de sua tupla têm valor (a constituição da tupla da EnO é definida na seção 3.5 e o significado dos demais símbolos podem ser obtidos na Lista de Símbolos que se encontra no início do texto). Esta primeira EnO é marcada com $t = 1$, isto é, o primeiro estado da sua história.

4.2.2 Eventos relacionados aos grupos

A criação de um novo grupo raiz a partir de uma especificação de grupo é formalmente detalhada a seguir:

Evento	criação de grupo raiz
Argumentos	gr : identificação do novo grupo gt : especificação do novo grupo
Pré-condições	$gt \in \mathcal{RG}$ (1) $gr \notin \mathcal{GI}$ (2)
Efeitos	$\mathcal{GI}^{t+1} = \mathcal{GI} \cup \{gr\}$ $grType^{t+1} = grType \cup \{gr \mapsto gt\}$

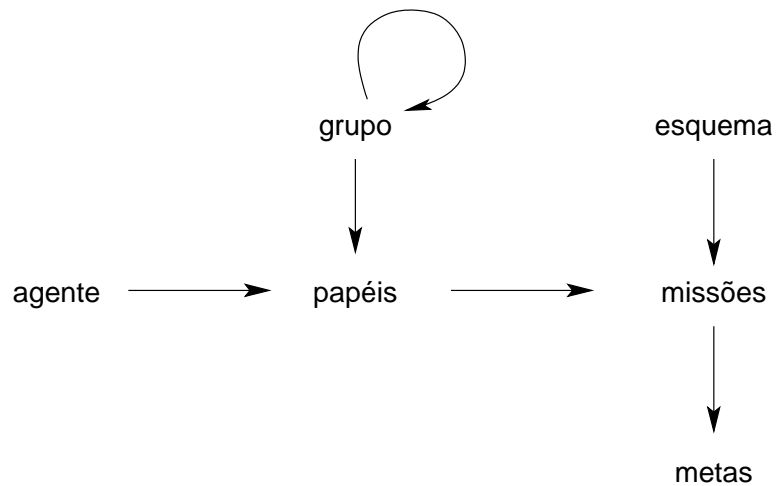
Este evento tem como pré-condições que a especificação de grupo gt do novo grupo gr pertença ao conjunto de especificações de grupo raiz (\mathcal{RG}), pré-condição (1), e que o novo grupo gr não exista, pré-condição (2). Como efeito deste evento, o conjunto \mathcal{GI} depois da ocorrência do evento (o estado do conjunto \mathcal{GI} no momento $t + 1$) será o conjunto \mathcal{GI} antes do evento (no momento t) mais a identificação do novo grupo. De forma semelhante, a função $grType$ é alterada para incluir o tipo deste novo grupo.

Do ponto de vista do funcionamento do OrgManager, a especificação acima indica que quando o OrgManager receber uma mensagem solicitando a criação de um grupo raiz, tal mensagem deve ser acompanhada de dois argumentos e semente será executada sobre a EnO se as pré-condições forem satisfeitas.

As restrições e efeitos da remoção de um grupo raiz são as seguintes:

Evento	remoção de grupo raiz
Argumentos	gr : identificação do grupo a ser removido
Pré-condições	$gr \in \mathcal{GI}$ (1) $isRootGr(grType(gr))$ (2) $\#grPlayers(gr, -) = 0$ (3) $\#subgroups(gr, -) = 0$ (4)
Efeitos	$\mathcal{GI}^{t+1} = \mathcal{GI} - \{gr\}$ $grType^{t+1} = grType - \{gr \mapsto grType(gr)\}$

As restrições para a remoção de um grupo raiz gr são: gr ser um grupo da EnO



Para remover um grupo, não deve haver agentes (com papéis) nele e nem sub-grupos. Logo, para remover um grupo deve-se remover primeiro todos os seus sub-grupos e agentes que desempenham papéis. Para um agente deixar um papel, precisa antes abandonar as missões desse papel. Por fim, para um agente abandonar uma missão, as metas desta missão ou devem ter sido satisfeitas ou serem impossíveis.

Figura 4.2: Dependência para remoção de um componente da EnO.

(pré-condição (1)) sua especificação de grupo ser raiz (pré-condição (2)) e não haver mais agentes (pré-condição (3)) ou sub-grupos nele (pré-condição (4)). A figura 4.2 apresenta, de forma mais global, a dependência entre os componentes da EnO no que concerne à sua remoção.

Evento	criação de grupo não raiz
Argumentos	gr : identificação do novo grupo gt : especificação do grupo sg : super-grupo (grupo que gr será sub-grupo)
Pré-condições	let $sgt = grType(sg)$, o tipo do super-grupo $sg \in \mathcal{GI}$ (1) $gr \notin \mathcal{GI}$ (2) $gt \in sgt.\mathcal{SG}$ (3) $\#subgroups(sg, gt) < second(sgt.ng(gt))$ (4)
Efeitos	$\mathcal{GI}^{t+1} = \mathcal{GI} \cup \{gr\}$ $grType^{t+1} = grType \cup \{gr \mapsto gt\}$ $subGr^{t+1} = subGr \oplus \{sg \mapsto (subGr(sg) \cup \{gr\})\}$

Basicamente, as restrições para a criação de um sub-grupo são: (1) o super-grupo existe, (2) o novo grupo não existe, (3) o tipo do novo grupo (gt) pertence ao conjunto dos tipos de sub-grupos do super-grupo e (4) a cardinalidade estabelecida para gt é respeitada (pré-condição 4: o número de sub-grupos de sg com tipo gt ainda é menor que o limite máximo de sub-grupos do tipo gt conforme a especificação do super-grupo sgt , ng é a função que retorna o número mínimo e o número máximo de sub-grupos de um tipo — ver página 45). O símbolo \oplus significa sobrescrita de função — ver a Lista de Símbolos.

A remoção de um grupo não raiz tem as seguintes pré-condições e efeitos:

Evento	remoção de grupo não raiz
Argumentos	gr : identificação do grupo
Pré-condições	$gr \in \mathcal{GI}$ (1) $\#grPlayers(gr, -) = 0$ (2) $\#subgroups(gr, -) = 0$ (3)
Efeitos	let $sg = superGr(gr)$, o super-grupo de gr $\mathcal{GI}^{t+1} = \mathcal{GI} - \{gr\}$ (1) $grType^{t+1} = grType - \{gr \mapsto grType(gr)\}$ (2) $subGr^{t+1} = subGr \oplus \{sg \mapsto (subGr(sg) - \{gr\})\}$ (3)

As restrições para a remoção de um sub-grupo são: (1) o grupo existe, (2) não há mais agentes no grupo e (3) nem sub-grupos. Os efeitos na EnO, caso este evento ocorra, são: (1) o conjunto de grupos não terá mais o grupo removido, (2) o mapeamento de tipos de grupo não terá mais no seu domínio o grupo removido e (3) o super-grupo não terá mais o sub-grupo removido.

4.2.3 Eventos relacionados aos esquemas

A criação de uma instância de esquema é feita por um agente que tem permissão para a meta raiz do esquema. Na sua criação, o esquema recebe um conjunto de grupos responsáveis pela execução do esquema, sendo que somente agentes pertencentes a estes grupos podem se comprometer com as missões do esquema. Mais precisamente, tem-se:

Evento	criação de esquema
Argumentos	si : identificação do novo esquema sch : especificação do esquema grs : conjunto de grupos responsáveis pelo esquema
Pré-condições	$sch \in \mathcal{S}_{fs}$ (1) $\forall gr \in grs \bullet gr \in \mathcal{GI}$ (2)
Efeitos	$\mathcal{SI}^{t+1} = \mathcal{SI} \cup \{si\}$ $scType^{t+1} = scType \cup \{si \mapsto (sch, grs)\}$ $gState^{t+1} = gState \cup$ $\bigcup_{g \in sch.G} (si, g) \mapsto (unsatisfied, uncommitted, forbidden)$

O último efeito indica o estado inicial (definido pela função $gState$) das metas do novo esquema.

Para o término normal de um esquema, não pode haver agentes comprometidos com as missões do esquema. Este evento também causa a atualização da taxa de sucesso deste tipo de esquema. Se a meta raiz do esquema está satisfeita, a taxa de sucesso aumenta, caso contrário, diminui. Assim como um esquema pode terminar normalmente, pode-se desistir da sua execução. Neste segundo caso, os agentes perdem seu compromisso com as missões do esquema. De forma análoga ao término normal de um esquema, a taxa de sucesso do esquema é atualizada. Estes dois eventos são detalhados a seguir.

Evento	término de um esquema
Argumentos	si : identificação do esquema
Pré-condições	$si \in \mathcal{SI}$ (1) $\#schPlayers(si, -) = 0$ (2)
Efeitos	$\mathcal{SI}^{t+1} = \mathcal{SI} - \{si\}$ $scType^{t+1} = scType - \{si \mapsto scType(si)\}$

Evento	esquema abortado
Argumentos	si : identificação do esquema
Pré-condições	$si \in \mathcal{SI}$ (1)
Efeitos	$\mathcal{SI}^{t+1} = \mathcal{SI} - \{si\}$ $scType^{t+1} = scType - \{si \mapsto scType(si)\}$ $\forall \alpha \in \mathcal{A} \forall m \in first(agMis(\alpha)) \bullet agMis^{t+1} =$ $agMis \oplus \{\alpha \mapsto agMis(\alpha) - \{(m, si)\}\}$

Sobre as metas de um esquema, os seguintes eventos podem ocorrer: a meta tornar-se satisfeita ou tornar-se impossível. A camada de aplicação é responsável por gerar os eventos que irão alterar o nível de satisfatibilidade de uma meta, porém a camada organizacional estabelece algumas restrições para a ocorrência destes eventos. Para uma meta de um esquema se tornar satisfeita, as pré-condições são as seguintes: ela deve ser permitida (conforme a definição da figura 4.3), ter pelo menos um agente comprometido com ela e ser possível. Para tornar-se impossível, como pré-condições tem-se que ela deve ser permitida e ter pelo menos um agente comprometido com ela. Segue a descrição detalhada dos eventos que alteram o estado de um meta.

Evento	meta foi satisfeita
Argumentos	si : identificação do esquema g : identificação da meta
Pré-condições	$si \in \mathcal{SI}$ (1) $g \in \mathcal{G}_{fs}$ (2) $isPermitted(si, g)$ (3) $isCommitted(si, g)$ (4) $isPossible(si, g)$ (5)
Efeitos	$\mathcal{SI}^{t+1} = \mathcal{SI} \cup \{si\}$ $gState^{t+1} =$ $gState \oplus \{(si, g) \mapsto (satisfied, committed, permitted)\}$

```

function isPermitted(esquema si, meta g)
if g é a raiz de si then
  return true
else
  g está em um plano do tipo “ $g_0 = \dots g \dots$ ”
  if g está em um plano do tipo “ $g_0 = \dots g_i , g \dots$ ” then
    if a meta  $g_i$  está satisfeita then                                %  $first(gState(si, g_i)) = satisfied$ 
      return true
    else
      return false
    end if
  else
    return isPermitted(si,  $g_0$ )
  end if
end if

function isPossible(esquema si, meta g)
if g não for impossível then                                       %  $first(gState(si, g)) \neq impossible$ 
  return true
else
  return false
end if

function isCommitted(esquema si, meta g)
if existir pelo menos um agente comprometido com g then
  %  $\exists \alpha \in \mathcal{A} \exists mp \in agMis(\alpha) \bullet second(mp) = si \wedge g \in schType(si).\mathcal{G}$ 
  return true
else
  return false
end if

```

Figura 4.3: Funções de verificação do estado de uma meta.

Evento	meta tornou-se impossível	
Argumentos	si : identificação do esquema g : identificação da meta	
Pré-condições	$si \in \mathcal{SI}$	(1)
	$g \in \mathcal{G}_{fs}$	(2)
	$isPermitted(si, g)$	(3)
	$isCommitted(si, g)$	(4)
Efeitos	$\mathcal{SI}^{t+1} = \mathcal{SI} \cup \{si\}$ $gState^{t+1} =$ $gState \oplus \{(si, g) \mapsto (impossible, committed, permitted)\}$	

Dos três argumentos que determinam o valor de uma meta em um esquema, só o primeiro (com valores *unsatisfied*, *satisfied* e *impossible*) é alterado por um evento organizacional sobre as metas. O segundo argumento (*committed* ou *uncommitted*) é automaticamente alterado pelo OrgManager quando um agente se compromete com uma missão que inclui a meta (ver função *isCommitted* na figura 4.3). De modo semelhante, o terceiro argumento (*permitted* ou *forbidden*) é alterado conforme as metas que são pré-condição vão sendo satisfeitas no esquema (ver função *isPermitted* na figura 4.3).

4.2.4 Eventos relacionados aos agentes

A arquitetura aqui proposta não restringe a entrada de agentes no sistema, as restrições são estabelecidas na adoção de papéis nos grupos. Outras propostas de arquitetura apresentam regras para a entrada e a saída de um agente: em (GLASER; MORIGNOT, 1997; FERBER; GUTKNECHT, 1998) algumas destas formas são apresentadas. Em geral, um agente pode entrar ou sair de uma sociedade se isso contribuir para a finalidade desta sociedade ou para os objetivos do agente³.

A entrada de um novo agente simplesmente o inclui no conjunto de agentes da EnO. Nenhuma restrição é estabelecida pela arquitetura.

³Conte e Castelfranchi (1995) explicam como um agente, mesmo autônomo, pode vir a assumir um papel, e as consequentes obrigações, em um sistema.

Evento	entrada de um novo agente na EnO
Argumentos	α : identificação do novo agente
Pré-condições	$\alpha \notin \mathcal{A}$ (1)
Efeitos	$\mathcal{A}^{t+1} = \mathcal{A} \cup \{\alpha\}$

A saída de um agente tem como pré-condições que este agente não esteja mais assumindo nenhum papel ou missão na EnO. Claro que este é o caso de uma saída *normal*, isto é, deixar a EnO de forma correta e conforme a organização social. Contudo, um agente autônomo pode deixar uma missão (ou papel) sem atender os requisitos listados aqui. Para estas *exceções* pode haver um sistema de penalidades no SMA. Este aspecto não é especificado no modelo \mathcal{MOISE}^+ .

Evento	saída de um agente da EnO
Argumentos	α : identificação do novo agente
Pré-condições	$\alpha \in \mathcal{A}$ (1)
	$\#agRole(\alpha) = 0$ (2)
	$\#agMis(\alpha) = 0$ (3)
Efeitos	$\mathcal{A}^{t+1} = \mathcal{A} - \{\alpha\}$

A adoção de um papel por um agente sempre ocorre em um grupo pois a noção de papel é relacional e, no modelo \mathcal{MOISE}^+ , é o papel que relaciona um agente a um grupo. As pré-condições para a adoção e abandono de um papel e seus efeitos na EnO são detalhadas pelos seguintes eventos:

Evento	adoção de um papel
Argumentos	α : identificação do agente ρ : identificação do novo papel gr : identificação do grupo onde o papel será assumido
Pré-condições	$\text{let } gt = grType(gr), \text{ o tipo do grupo de } gr$ $\alpha \in \mathcal{A}$ (1) $\rho \in gt.\mathcal{R}$ (2) $\#grPlayers(gr, \rho) < second(gt.np(\rho))$ (3) $\forall \rho_b \in \mathcal{R} \bullet$ (4) $(\rho_b, gr) \in agRole(\alpha) \Rightarrow \rho \bowtie^{intra} \rho_b$ $\forall \rho_c \in \mathcal{R}_{ss} \bullet$ (5) $((\rho_c, gr') \in agRole(\alpha) \wedge gr \neq gr') \Rightarrow \rho \bowtie^{inter} \rho_c$
Efeitos	$agRole^{t+1} = agRole \oplus \{\alpha \mapsto (agRole(\alpha) \cup \{(\rho, gr)\})\}$

Para um agente α assumir um papel ρ no grupo gr , as seguintes pré-condições devem ser satisfeitas: (1) o agente deve pertencer a EnO; (2) o papel que irá assumir deve pertencer ao conjunto de papéis da especificação do grupo gr ; (3) o número de agentes com tal papel deve ser menor que o número máximo permitido pela cardinalidade do papel no grupo; (4) para todos os papéis que o agente tem no grupo gr , o novo papel deve ser intra-grupo compatível; e (5) para todos os papéis que o agente tem em outros grupos que não gr , deve haver uma compatibilidade inter-grupo.

Evento	abandono de um papel
Argumentos	α : identificação do agente ρ : identificação do papel gr : identificação do grupo
Pré-condições	$(\rho, gr) \in agRole(\alpha)$ (1) $\neg \exists m \in first(agMis(\alpha)) \bullet per(\rho, m, tc) \wedge$ (2) $gr \in second(scType(m))$
Efeitos	$agRole^{t+1} = agRole \oplus \{\alpha \mapsto (agRole(\alpha) - \{(\rho, gr)\})\}$

Para deixar de desempenhar um papel, um agente deve já ter assumido previamente tal papel e não deve existir uma missão que tenha sido assumida por meio do papel que se quer deixar. Novamente, especificou-se o abandono *normal* de um papel. O abandono de um papel sem respeitar estas condições caracteriza uma exceção.

A relação dos agentes com os esquemas é dada pelas missões. Dois eventos organizacionais alteram esta relação: comprometimento e descomprometimento com uma missão.

Evento	comprometimento com uma missão
Argumentos	α : identificação do agente m : identificação da missão si : identificação do esquema
Pré-condições	$\text{let } sch = \text{first}(scType(si)), \text{ o tipo do esquema } si$ $\alpha \in \mathcal{A}$ (1) $\#schPlayers(si, m) < \text{second}(sch.nm(m))$ (2) $gState(\text{getRootGoal}(si)) = \text{unsatisfied}$ (3) $\exists(\rho, gr) \in agRole(\alpha) \bullet \text{per}(\rho, m, tc) \wedge$ (4) $gr \in \text{second}(scType(si)) \wedge$ $now \in tc$
Efeitos	$agMis^{t+1} = agMis \oplus \{\alpha \mapsto (agMis(\alpha) \cup \{(m, si)\})\}$

Para um agente se comprometer com uma missão, as seguintes pré-condições devem ser satisfeitas: (1) ele deve estar na EnO; (2) o número máximo de agentes comprometidos com a missão ainda não ter sido atingido; (3) o esquema não deve ter terminado (termina quando a meta raiz é satisfeita ou impossível); e (4) pelo menos um dos papéis do agente ter permissão para a missão, sendo que este papel deve ter sido assumido dentro dos grupos responsáveis pelo esquema ($\text{second}(scType(si))$).

Evento	descomprometimento com uma missão	
Argumentos	α : identificação do agente m : identificação da missão si : identificação do esquema	
Pré-condições	$(m, si) \in agMis(\alpha)$	(1)
	$\forall g \in mo(m) \bullet$	(2)
	$(first(gState(si, g)) = satisfied) \vee$	
	$(first(gState(g, si)) = impossible)$	
Efeitos	$agMis^{t+1} = agMis \oplus \{\alpha \mapsto agMis(\alpha) - \{(m, si)\}\}$	

Para um agente remover seu compromisso com uma missão ele deve já ter previamente se comprometido com ela (1) e todas as metas desta missão devem ou ter sido satisfeitas ou serem impossíveis (2).

Tendo-se o detalhamento das condições para a ocorrência dos eventos que alteram o estado da EnO mantida pelo OrgManager, a próxima seção descreve como os agentes geram estes eventos e se mantém informados do estado da EnO.

4.3 Componente OrgBox

Os agentes têm, por meio do OrgBox, uma visão local da EnO que é mantida pelo OrgManager. O OrgBox tem as seguintes funções na arquitetura: garantir o cumprimento das restrições organizacionais e prover alguns serviços à arquitetura do agente. A arquitetura do MOISE⁺ termina neste componente (conforme ilustra a figura 4.1) para manter a propriedade de independência em uma arquitetura de camadas (uma camada só depende da camada inferior). Portanto, não são feitas considerações sobre como é a arquitetura de funcionamento dos agentes da camada de aplicação. Por outro lado, a arquitetura dos agentes, na camada de aplicação, utiliza e depende do OrgBox como forma de se situar organizacionalmente.⁴ Assim, via OrgBox, vários tipos de arquiteturas de agente podem

⁴Um exemplo de uma possível arquitetura de agente organizacional, baseado na arquitetura ASIC (BOISSIER; DEMAZEAU, 1994; BOISSIER, 1993) é proposta por Hannoun (2002). No estudo de caso do capítulo 8, também é apresentada uma versão bastante simplificada de arquitetura de agente.

fazer uso da camada organizacional do *MOISE*⁺.

A visão local que o *OrgBox* tem da *EnO* não é exatamente uma cópia da *EnO* mantida pelo *OrgManager*. Por causa das ligações do tipo *acquaintance*, o conjunto de agentes na visão local é formado apenas pelos agentes que possuem uma ligação de conhecimento com o agente do *OrgBox*. Outro aspecto importante é a atualização da visão local em relação ao estado real da *EnO* mantida pelo *OrgManager*. A solução proposta é o *OrgManager* notificar os *OrgBoxes* das mudanças no estado da *EnO* quanto ocorrem atualizações pertinentes ao *OrgBox*. Por exemplo, se uma meta tornou-se impossível, o *OrgBox* será informado somente se participar do esquema que contém tal meta. Assim as visões locais não estão sempre completamente sincronizadas com a *EnO*, mas estão suficientemente atualizadas.

Os principais serviços que o *OrgBox* oferece à arquitetura do agente são os seguintes.

Comunicação: o envio e recebimento de mensagens KQML para/de outros agentes. Neste serviço, o *OrgBox* implementa as restrições das ligações de comunicação de modo que o agente somente enviará/receberá mensagens dos agentes para os quais possua uma ligação de comunicação.⁵ Um agente *ag*₁ possui uma ligação de comunicação com outro agente *ag*₂ se um de seus papéis possuir uma ligação de comunicação com um dos papéis do agente *ag*₂.

Geração de eventos organizacionais: por meio do *OrgBox*, o agente pode entrar no sistema, assumir um papel, criar um grupo, etc. O *OrgBox* encapsula a comunicação KQML necessária para gerar os eventos organizacionais que o *OrgManager* compreende.

Informações de obrigações: o *OrgBox* mantém o agente informado de quais missões ele é obrigado a se comprometer (o algoritmo da figura 4.4 detalha a obtenção destas missões — este algoritmo já ordena as missões pela ordem de preferência estabelecida na EF). Estas missões dependem essencialmente

⁵Certamente um agente pode burlar este mecanismo de controle bastando encontrar outro meio de comunicação com os demais agentes (utilizando sockets, por exemplo). Contudo não é foco deste trabalho analisar as transgressões organizacionais.

```

function getObligatedMissions(agent  $\alpha$ )

 $all \leftarrow \langle \rangle$                                      % a lista de missões obrigatórias
                                                         %  $\langle \rangle$  representa uma lista vazia

for all papel  $\rho$  que  $\alpha$  assumiu do                 %  $\rho \in first(agRole(\alpha))$ 
   $gr \leftarrow$  o grupo onde  $\rho$  foi assumido
  for all esquema  $si$  que  $gr$  é responsável do
    if o esquema ainda não terminou then             %  $getRootGoal(si) = unsatisfied$ 
      for all missão  $m$  do esquema  $si$  do             %  $m \in schType(si).M$ 
        if  $\rho$  tem obrigação para com  $m$  then       %  $obl(\rho, m, -) \in \mathcal{O}$ 
          if cardinalidade da missão  $m$  não está ok then
            %  $first(schType(si).nm(m)) < \#schPlayers(si, m)$ 
             $all \leftarrow append(all, \langle m \rangle)$ 
          end if
        end if
      end for
    end if
  end for
end for
ordena  $all$  de acordo com as preferências entre as missões
return  $all$ 

```

Figura 4.4: Algoritmo que determina as obrigações de um agente.

dos papéis que o agente assumiu e das obrigações (determinadas na ED) decorrentes destes papéis.

Nos casos em que as obrigações de um agente não estejam incluídas nos seus comprometimentos ($getObligatedMissions(\alpha) \not\subseteq agMis(\alpha)$), diz-se que o agente não está cumprindo com suas obrigações organizacionais.

Informações de metas possíveis: o OrgBox mantém o agente informado de quais são as metas globais que podem ser buscadas (o algoritmo da figura 4.5 detalha a obtenção destas metas). O conjunto destas metas depende do estado de execução dos esquemas e depende, portanto, das ações dos outros agentes.

O OrgBox somente informa quais são as metas possíveis, sendo que o agente deve assumir a responsabilidade pela sua satisfação, inclusive pela construção de um plano para satisfazê-la caso não haja um plano no esquema. O que se estabelece é a responsabilidade pela meta, para satisfazer a meta o agente pode, inclusive, delegar a tarefa a outro agente. Portanto, apesar da organização indicar quais são as metas globais dos agentes, eles têm

```

function getPossibleGoals(agent  $\alpha$ )

   $all \leftarrow \langle \rangle$                                      % a lista de metas possíveis
                                                         %  $\langle \rangle$  representa uma lista vazia
  for all missão  $m$  que  $\alpha$  está comprometido do      %  $m \in first(agMis(\alpha))$ 
     $sch \leftarrow$  o tipo do esquema onde  $m$  foi assumido
    for all meta  $g$  da missão  $m$  do                    %  $g \in sch.mo(m)$ 
      if  $\neg isSatisfied(g) \wedge isPossible(g) \wedge isPermitted(g)$  then
        for all  $g_s$  que é super-meta no  $sch$  do
          if  $\neg isSatisfied(g_s) \wedge \neg isImpossible(g_s)$  then
             $all \leftarrow append(all, \langle g \rangle)$ 
          end if
        end for
      end if
    end for
  end for
  ordena  $all$  de acordo com as preferências entre as missões
  return  $all$ 

```

A satisfação de uma meta somente pode ser buscada por um agente se tal meta estiver na lista retornada pela função acima. De forma geral, o algoritmo especifica que para uma meta pertencer a esta lista, o agente deve estar comprometido com a missão desta meta, a meta não deve estar satisfeita, a meta deve ser possível, a meta deve ser permitida pelo estado de execução do esquema (ver algoritmo da figura 4.3) e nenhuma super-meta deve estar satisfeita ou impossível. Esta última condição justifica-se pelo fato de que se uma super-meta já estiver satisfeita (ou impossível) é inútil buscar satisfazer suas sub-metas. A lista all depende das ações dos demais agentes pois eles podem alterar o estado das metas do esquema e conseqüentemente o valor retornado pela função $isPermitted$. Por fim, a lista de metas possíveis é ordenada segundo a ordem de preferência entre as missões destas metas.

Figura 4.5: Algoritmo que determina as metas possíveis para um agente.

autonomia de planejamento.

4.4 Exemplo de utilização da arquitetura

Para exemplificar o funcionamento da arquitetura e como ocorre a coordenação entre os agentes envolvidos na execução de um esquema, será considerado o seguinte contexto:

- A EO construída para a escola nos exemplos 3.1 (página 48), 3.5 (página 63) e 3.7 (página 69). O anexo B.1 tem a descrição completa desta EO e a

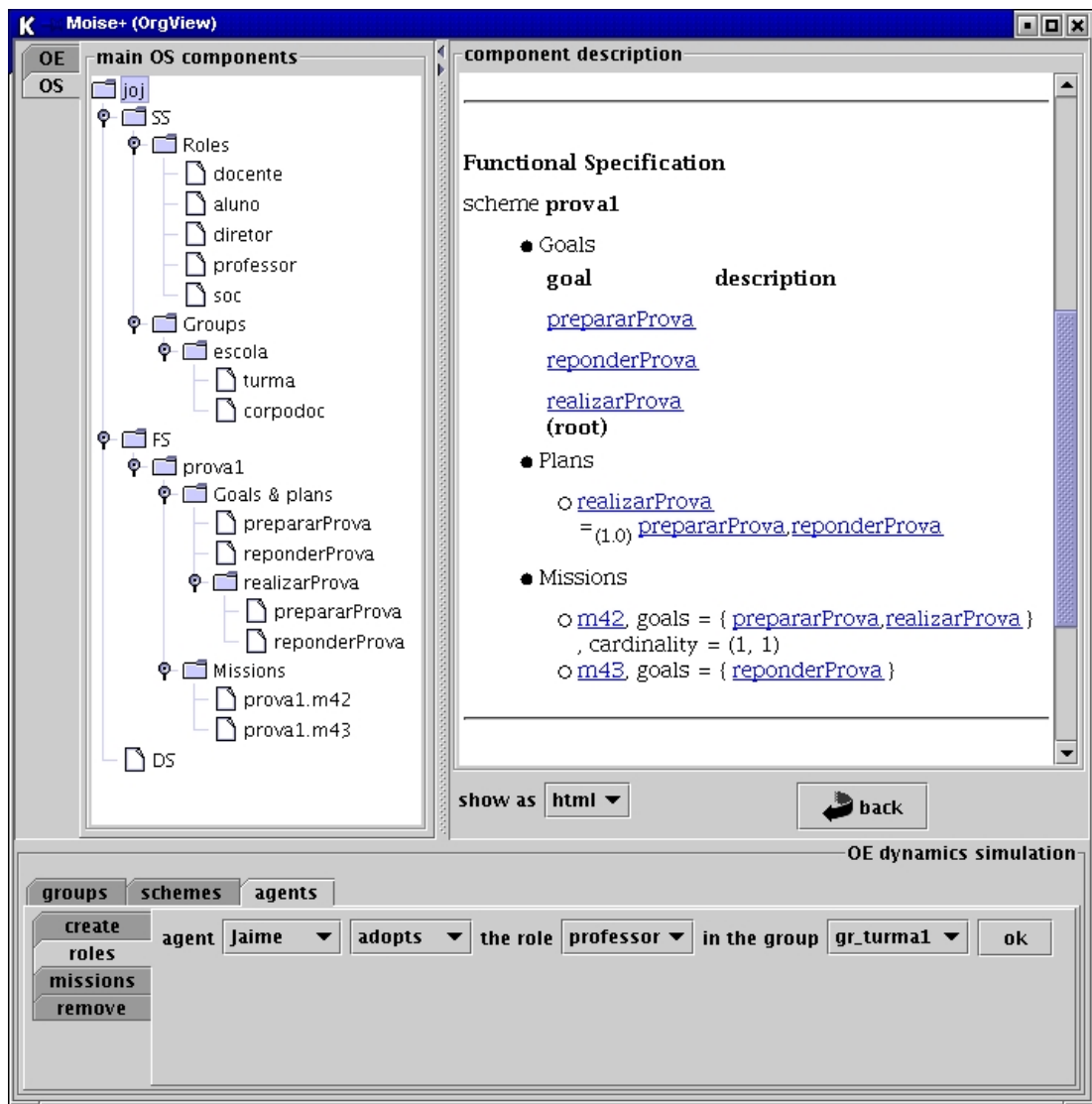


Figura 4.6: Especificação organizacional da escola.

figura 4.6 a resume. As figuras de telas que seguem foram obtidas a partir da implementação desta arquitetura.

- A EnO criada possui uma turma com um agente professor (chamado Jaime) e três alunos (chamados Jomi, Gustavo e Fabrício). A figura 4.7 ilustra o estado deste grupo.

A partir deste contexto, este exemplo irá detalhar os acontecimentos que ocorrem na camada organizacional deixando de lado a camada de aplicação e não detalhando como os agentes funcionam. Particularmente, será exemplificado como ocorre a execução do esquema de realização de uma prova através dos passos

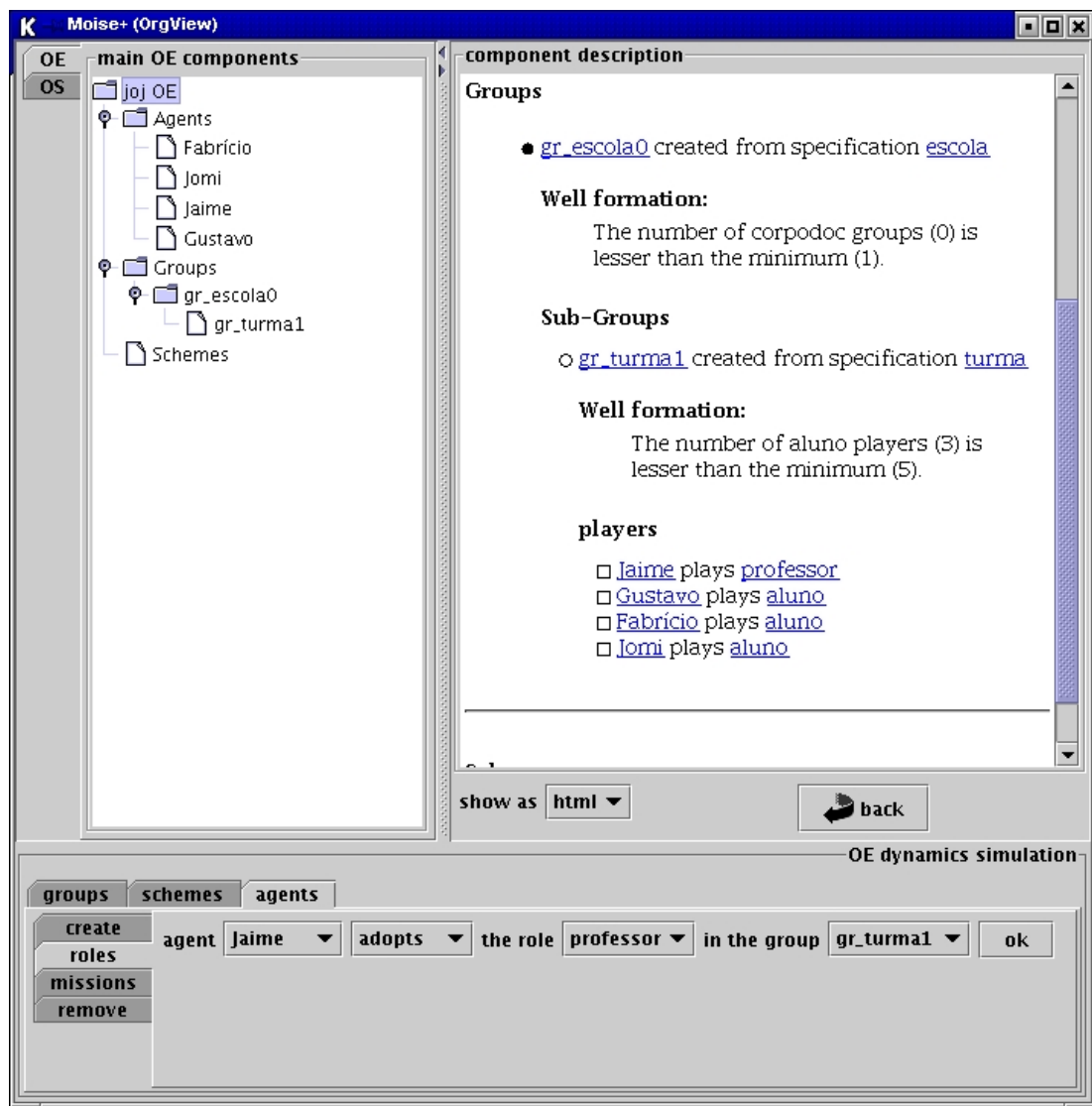


Figura 4.7: Especificação dos grupos da escola.

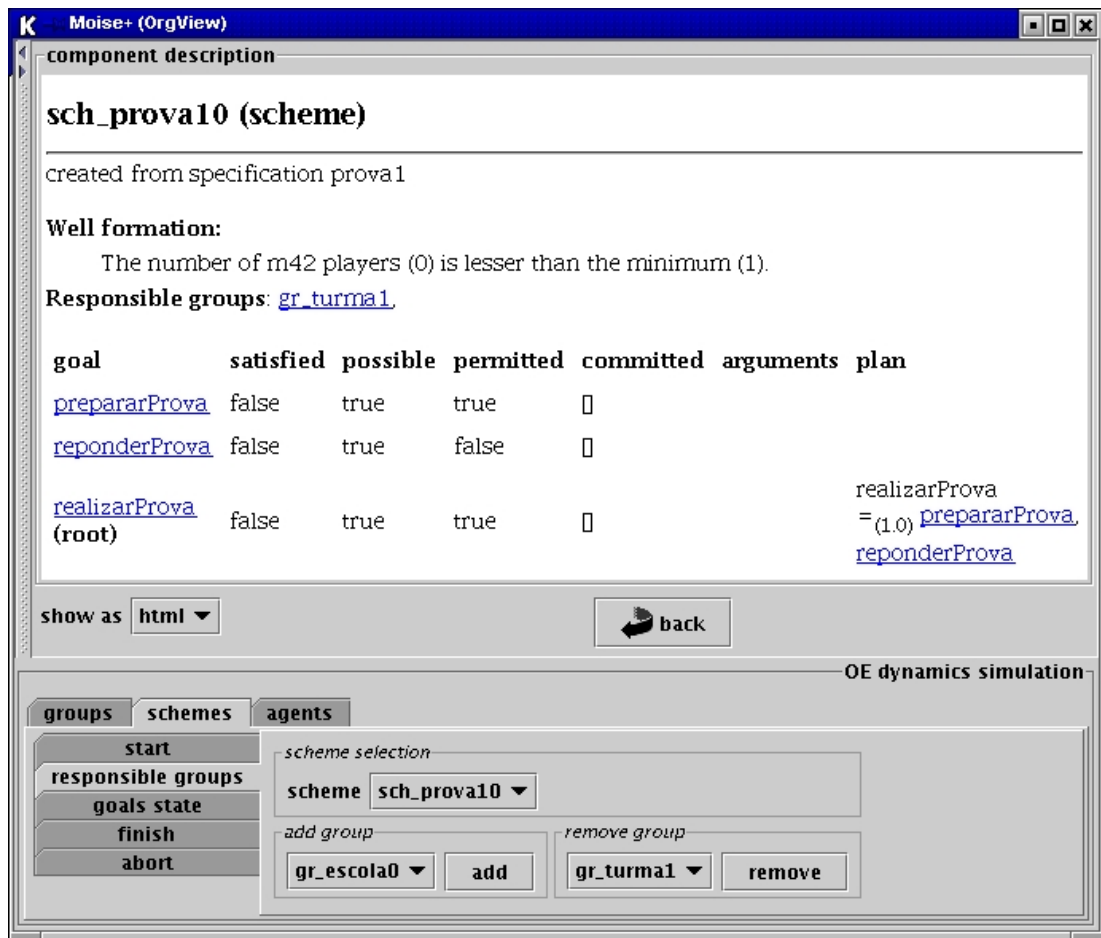


Figura 4.8: Estado inicial do esquema de realização de uma prova.

seguintes.

1. O agente Jaime, por meio de seu OrgBox, solicita a criação de uma instância do esquema *prova1*. Como este agente atende às pré-condições de criação do esquema, ele tem permissão para a meta “realizarProva” que é a raiz do esquema e o esquema é criado. O estado do esquema após sua criação é ilustrado na figura 4.8. Note que o esquema não está bem formado já que não há nenhum agente assumindo a missão m_{42} (realizar e preparar a prova). As metas não estão satisfeitas, todas são possíveis, nenhuma tem agentes comprometidos e somente duas são permitidas. A meta “reponderProva” somente pode ser executada quando a meta “prepararProva” já estiver satisfeita (veja o plano para a meta “realizarProva”).
2. O agente Jaime, através da invocação da função *getObligatedMissions* do

seu OrgBox, descobre que deveria se comprometer com a missão m_{42} . A arquitetura do agente Jaime decide que ele deve se comprometer com tal missão. O pedido é feito, via OrgBox, ao OrgManager que altera o estado do esquema como tendo Jaime comprometido com m_{42} .

3. Os três alunos, de forma análoga, percebem que deveriam se comprometer com a missão m_{43} (responder a prova) e se comprometem com tal missão. O novo estado do esquema está ilustrado na figura 4.9.
4. O agente Jaime consulta no OrgBox (função *getPossibleGoals*) as metas globais que lhe são possíveis. Dado que “prepararProva” e “realizarProva” são permitidas (conforme figura 4.9) e não há uma relação de preferência entre elas já que pertencem à mesma missão, o OrgBox retorna a meta mais a esquerda na árvore do esquema, isto é, a meta “prepararProva”.
5. Os alunos fazem o mesmo, mas não existem metas possíveis para eles. Apesar de comprometidos com a meta “responderProva” tal meta não é permitida (ver figura 4.9).
6. O agente Jaime termina a preparação da prova e notifica o OrgManager que a meta “realizarProva” foi satisfeita. A tela da figura 4.10 ilustra o novo estado do esquema. Nota-se que a meta “realizarProva” tornou-se permitida.
7. Os alunos consultam novamente suas metas possíveis e percebem que agora a meta “responderProva” é permitida e passam a executá-la. Ao final da prova, estes agentes tornam a meta “responderProva” satisfeita.

Percebe-se aqui que a camada organizacional facilitou a coordenação entre os agentes dispensando-os de se comunicarem para realizarem suas ações na seqüência correta.
8. O agente Jaime, percebendo que o plano para a meta “prepararProva” foi executado, torna a meta “prepararProva” satisfeita. Como esta é a meta raiz do esquema, o esquema é considerado como terminado.
9. Não tendo mais metas possíveis para a missão m_{43} , os alunos retiram seu comprometimento para com esta missão.

component description

sch_prova10 (scheme)

created from specification prova1

Well formation:
ok

Responsible groups: [gr_turma1](#).

players

- [Jaime](#) committed to [m42](#)
- [Fabricio](#) committed to [m43](#)
- [Jomi](#) committed to [m43](#)
- [Gustavo](#) committed to [m43](#)

goal	satisfied	possible	permitted	committed	arguments	plan
prepararProva	false	true	true	[Jaime]		
reponderProva	false	true	false	[Fabricio, Gustavo, Jomi]		
realizarProva (root)	false	true	true	[Jaime]		realizarProva =(1.0) prepararProva , reponderProva

show as

OE dynamics simulation

groups | **schemes** | agents

create roles missions remove

agent [Jomi](#) commits to the mission [prova1.m43](#) in the scheme [sch_prova10](#) ok

Figura 4.9: Estado do esquema de realização de uma prova após os comprometimentos dos agentes.

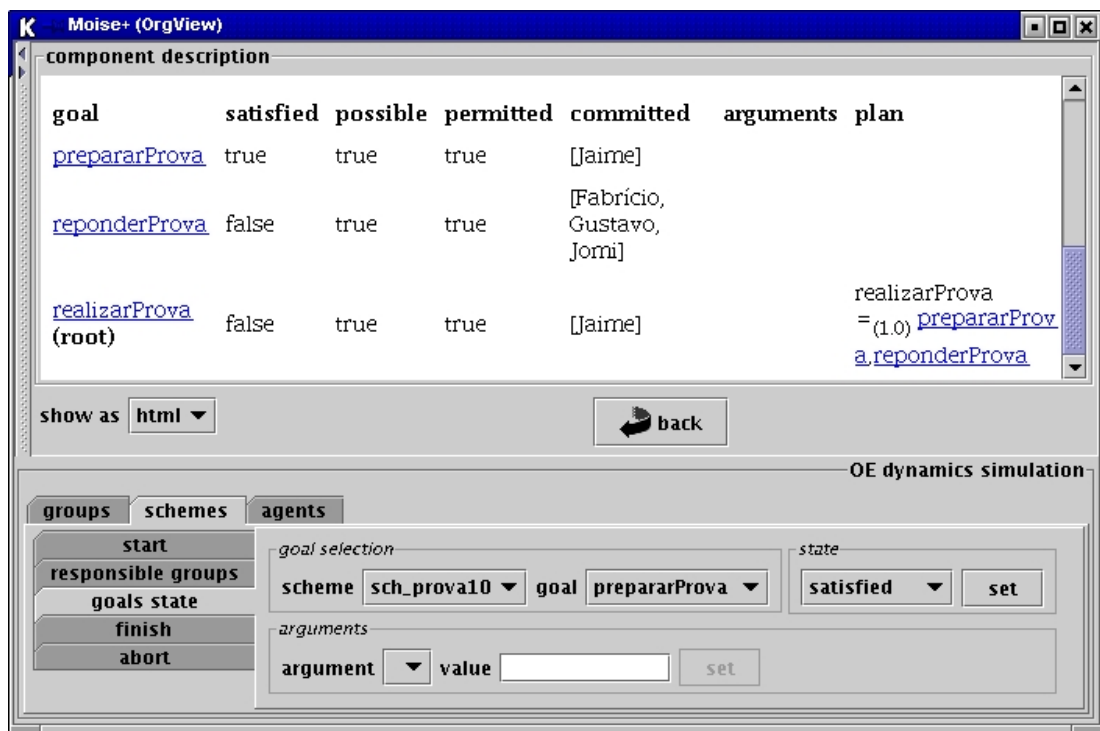


Figura 4.10: Estado do esquema de realização de uma prova após a realização da meta *preparar prova*.

10. Analogamente, o professor retira seu comprometimento para com as missões do esquema encerrado. A tela da figura 4.11 ilustra o estado final do esquema.

4.5 Conclusões

Ao contrário da maioria das soluções de implementação de modelos organizacionais onde o modelo está *dentro* dos agentes, como por exemplo (HANNOUN, 2002; WERNER, 1989; HORLING; BENYO; LESSER, 2001; STONE; VELOSO, 1999), a arquitetura proposta busca considerar sistemas abertos. Em tais sistemas não se conhece a arquitetura dos agentes que participam no sistema. Por esta razão, a proposta apresentada faz poucas exigências quanto à arquitetura dos agentes, fazendo uso de uma abordagem em camadas. A única exigência para um agente participar do sistema é ele conhecer KQML (para se comunicar com o OrgManager) ou utilizar o OrgBox (que encapsula a comunicação KQML).

Tal solução permitiu garantir o cumprimento das ligações de conhecimento

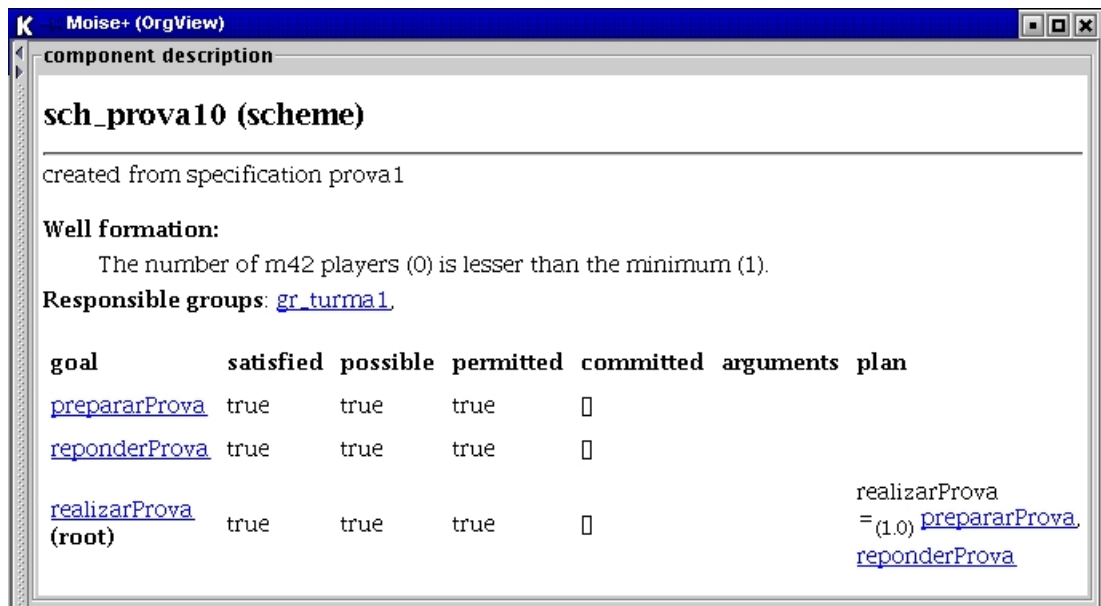


Figura 4.11: Estado final do esquema de realização de uma prova.

e de comunicação do modelo \mathcal{MOISE}^+ : a arquitetura só permite a comunicação entre agentes com papéis que tenham uma ligação de comunicação e um agente somente pode conhecer aqueles que tenham ligações de conhecimento com ele. Além de garantir o cumprimento destes dois tipos de ligação, a arquitetura proposta viabiliza a coordenação na execução dos objetivos que estão distribuídos entre os agentes comprometidos com um esquema. Por outro lado, a ligação de autoridade não é verificada na camada organizacional e deve ser implementada na arquitetura do agente (camada de aplicação). Apesar do OrgBox informar ao agente quais suas obrigações (missões e metas), ele não garante que o agente irá satisfazê-las. Novamente, depende da arquitetura do agente a decisão de satisfazer as metas globais.

Parte II

Reorganização

5 Reorganização de Sistemas Multiagentes

Se, por um lado, o estabelecimento de uma organização social melhora o desempenho de uma sociedade de agentes para uma finalidade, por outro lado, sua perenização diminui a flexibilidade desta sociedade, uma vez que se perde a capacidade de adaptação a novos problemas e situações, uma propriedade importante nos SMA (ALVARES; SICHMAN, 1997). Portanto, os SMA precisam ter a capacidade de alterar sua organização a fim de se adaptar a mudanças no seu ambiente ou mesmo para otimizar seu funcionamento.

Este capítulo apresenta e caracteriza o processo de reorganização de um SMA a partir de um ponto de vista coerente com a visão organizacional adotada (capítulo 3). Segundo tal ponto de vista, a solução que alguns pesquisadores deram para o problema de mudança organizacional é apresentada.

5.1 Introdução

Como visto nos capítulos precedentes, nas visões centradas na organização, a organização normalmente é construída sobre dois conceitos chave: *(i)* a especificação (papéis, especificação de grupos, ligações, planos globais,...) e *(ii)* o estado corrente de uma instância desta especificação formada por um conjunto de agentes com uma finalidade. Estes dois conceitos são, respectivamente, a EO e a EnO do modelo *MOISE*⁺. Tal visão permite definir, de forma geral, *reorganização* como um processo que altera o estado corrente destes dois aspectos.

Em cada um destes dois tipos mais gerais de mudança há uma grande variedade de possibilidades. A mudança pode ser, por exemplo, a simples adoção

de um papel, ou a remoção de um tipo de grupo. Para melhor exemplificar esta variedade de abordagens, na tabela 5.1 estão relacionados alguns trabalhos sobre reorganização e quatro aspectos são considerados: o modelo organizacional utilizado na reorganização, o que a reorganização muda na organização, como ocorre esta mudança e quem é responsável por conduzir a reorganização. Cada um destes aspectos é detalhado e exemplificado a seguir.

O modelo organizacional utilizado, embora nem sempre esteja explícito, determina várias propriedades do processo de reorganização. Por exemplo, não é possível alterar o agrupamento de agentes se o modelo organizacional for o TÆMS, pois a noção de grupo não existe no modelo; ou ainda, não é possível alterar a organização para uma forma hierárquica se não existe a noção de autoridade no modelo.¹

Quanto ao objeto da mudança, os processos de reorganização podem ser classificados em dois grandes tipos:

Nível entidade organizacional (EnO) : compreende mudanças nas missões e papéis que os agentes estão comprometidos, a criação de grupos e esquemas, etc. No caso do modelo $\mathcal{M}OISE^+$, este tipo de mudança faz parte da dinâmica da EnO, já descrita na seção 4.2.

Por exemplo, em (GLASER; MORIGNOT, 1997), o problema da entrada de um novo agente na sociedade é resolvido pela avaliação da utilidade dos papéis que o agente se dispõe a assumir. Basicamente, o agente é aceito se a utilidades dos papéis que irá assumir é maior que o custo de ter o agente desempenhando estes papéis. Dependendo do papel e da situação, ter um agente assumindo tal papel pode ser vantajoso ou não para o sistema. Ao final deste processo de decisão, o conjunto de agentes pode mudar, mas o conjunto de papéis conhecidos continua o mesmo independente do resultado da decisão, ou seja, apenas a EnO mudou e não a EO.

Em (ISHIDA; GASSER; YOKOO, 1992), um agente que desempenha vários papéis pode decidir repartir estes papéis em dois agentes caso não esteja conseguindo atingir os prazos estabelecidos; ou os papéis que dois agentes

¹Para simplificar a leitura do texto, será utilizada terminologia do $\mathcal{M}OISE^+$ para falar dos outros modelos organizacionais quando houver termos com significado equivalentes. Por exemplo, vários modelos distinguem EnO de EO, porém utilizam outros termos.

assumem podem ser atribuídos a um único agente caso haja necessidade de economizar recursos. Novamente, a EnO (o número de agentes) foi alterado e não a EO (o conjunto de papéis, neste caso).

Nível especificação organizacional (EO) : compreende mudanças na especificação organizacional. Neste nível, o \mathcal{MOISE}^+ permite considerar três sub-níveis de reorganização:

- mudanças na estrutura (EE: grupos, papéis, ligações, etc.),
- mudanças no funcionamento (EF: os esquemas, missões, preferências, etc.), e
- mudanças nas relações deônticas (ED: obrigações e permissões dos papéis).

Por exemplo, em (STRUGEON, 1995), os agentes podem escolher entre quatro tipos de especificação: mercado, hierarquia, comunidade ou sociedade (mudança no nível EE). Em (HORLING; BENYO; LESSER, 2001), a EO é descrita em TÆMS, logo, os agentes podem mudar as descrições de tarefas (mudança no nível EF). Em (SO; DURFEE, 1993), a EO é considerada como uma árvore de decomposição de tarefas, a reorganização é, portanto, uma alteração no fator de ramificação desta árvore (mudança no nível EF). Em (STONE; VELOSO, 1999), a EO representa a formação de um time de futebol com papéis sendo posições na formação; os jogadores alteram a formação (nível EE) para melhor se adaptar ao oponente.

Evidentemente, uma mudança no nível EO causa mudanças no nível EnO, já que este segundo depende do primeiro.

Enquanto há dois tipos de objeto de mudança para o processo de reorganização (o que muda), a forma como esta mudança se passa no decorrer do funcionamento do SMA também difere (como muda). Três formas podem ser concebidas:

Pré-definida: a reorganização já está prevista na especificação organizacional.

Por exemplo, em (STONE; VELOSO, 1999), os agentes podem previamente concordar que irão mudar a formação do time aos trinta minutos do segundo

Tabela 5.1: Classificação de alguns trabalhos sobre reorganização de SMA

Trabalho	<i>Modelo Organizacional</i>	<i>O que muda</i>	<i>Como muda</i>	<i>Quem muda</i>
Glaser e Morignot (1997)	A EO é definida por um conjunto de papéis (definidos como um conjunto de competências. A EnO é representada por pares (agente × papel)	Nível EnO: o conjunto de agentes incluindo os seus papéis	Controlado pela relação entre os papéis esperados pela sociedade e os papéis ofertados pelo novo agente	Endógena centralizada: o novo agente, que propõe à sociedade um conjunto de papéis
Ishida, Gasser e Yokoo (1992)	Alocação de papéis (na forma de tarefas) aos agentes	Nível EnO: os papéis alocados a cada agente	Controlado pelos requisitos de tempo de resposta e consumo de recursos	Endógena distribuída: cada agente toma ele mesmo a decisão de dividir-se em dois ou unir-se a outro agente
Strugeon (1995)	Quatro tipos de EE: mercado, hierarquia, comunidade ou sociedade	Nível EE: o tipo de organização	Controlado pelas características das tarefas que surgem para o SMA	Endógena centralizada: o agente que recebe a nova tarefa escolhe a organização e a implementa
Horling, Benyo e Lesser (2001)	A EO é definida em TEMS	Nível EF: a visão subjetiva que os agentes têm das tarefas	Controlado por um sistema de diagnóstico e correção de falhas	Endógena descentralizada: o sistema de controle (diagnóstico e correção) realiza as mudanças
So e Durfee (1993)	A organização é considerada como uma árvore de decomposição de tarefas	Nível EF: o fator de ramificação da tarefa e conseqüentemente sua alocação aos agentes	Controlado pelo cálculo do melhor fator de ramificação dada as propriedades da tarefas (tamanho, por exemplo)	Endógena centralizada: um gerente de execução monitora constantemente o sistema para identificar quando mudar a organização
Stone e Veloso (1999)	A organização possui uma EE (a formação do time) e uma EF (os comportamentos esperados de cada papel)	Nível EE: mudança na formação, da qual decorre mudança nos comportamentos esperados (nível EF)	Pré-definido por um acordo entre os agentes antes do jogo ou Controlado pela identificação de oportunidades no jogo	Endógena centralizada no caso de mudança por parte de um jogador que identifica a oportunidade para uma formação ou descentralizada no caso do acordo pré-definido

tempo. No modelo organizacional de Carron e Boissier (2001), restrições temporais previamente definidas podem determinar que um papel irá deixar de existir depois de um certo tempo e que logo em seguida um novo papel passará a existir.

Controlada: apesar de não se conhecer previamente quando a reorganização irá acontecer, o *processo* que altera a organização é conhecido. No caso de Horling, Benyo e Lesser (2001), o processo de alteração é determinado por um sistema de diagnóstico. Sabe-se, portanto, como acontece a reorganização. Obviamente esta normalização da reorganização pode se dar de diversas formas. Por exemplo, pode-se estabelecer que a reorganização deve se dar por meio de negociação ou, de forma bem distinta, pela autoridade dada a um agente para reorganizar. Em todos estes casos, está previamente estabelecido como será a reorganização.

Emergente: não há qualquer controle explícito por parte do sistema sobre o processo de reorganização. Este tipo de reorganização acontece, por exemplo, por iniciativa de um agente. Apesar de não haver uma especificação que determina como se reorganizar, isto é, a reorganização não está institucionalizada, um agente pode tomar a iniciativa de tentar mudar a organização segundo critérios e métodos dele mesmo. Não há, portanto, garantias de que sua iniciativa tenha sucesso, já que depende da aceitação dos demais agentes.²

No caso da reorganização controlada, em geral, o processo de reorganização é dividido nas seguintes etapas³: monitoração, projeto, seleção e implementação. Cada etapa, por sua vez, pode ser executada de duas maneiras (quem muda):

- i) *endógena*: um agente (abordagem centralizada) ou vários agentes (abordagem descentralizada) do sistema realizam a etapa (ver, por exemplo, a tabela 5.1). O termo *auto-organização* também é empregado para esse tipo

²Esta descrição de reorganização emergente considera uma sociedade do Tipo OC onde emerge um processo de reorganização sobre uma organização institucionalizada (explicitamente definida). O termo “reorganização emergente” não se aplica a organizações do Tipo AC pois, se não há uma organização explícita, seria muito difícil diferenciar *organização* emergente de *reorganização* emergente. Em outras palavras, seria possível distinguir se o que se observa do sistema é uma organização emergente ou uma reorganização emergente?

³Estas etapas são utilizadas, por exemplo, por Pattison, Corkill e Lesser (1987), So e Durfee (1993).

de reorganização. Normalmente, este tipo de reorganização é feita por agentes com capacidade de representação de sua organização (organizações do Tipo OC ou AC);

- ii) *exógena*: o usuário do SMA executa a etapa (por exemplo, Malone (1999), Tambe, Pynadath e Chauvat (2001)).

Na seqüência, cada uma destas etapas serão detalhadas.

5.2 Monitoração

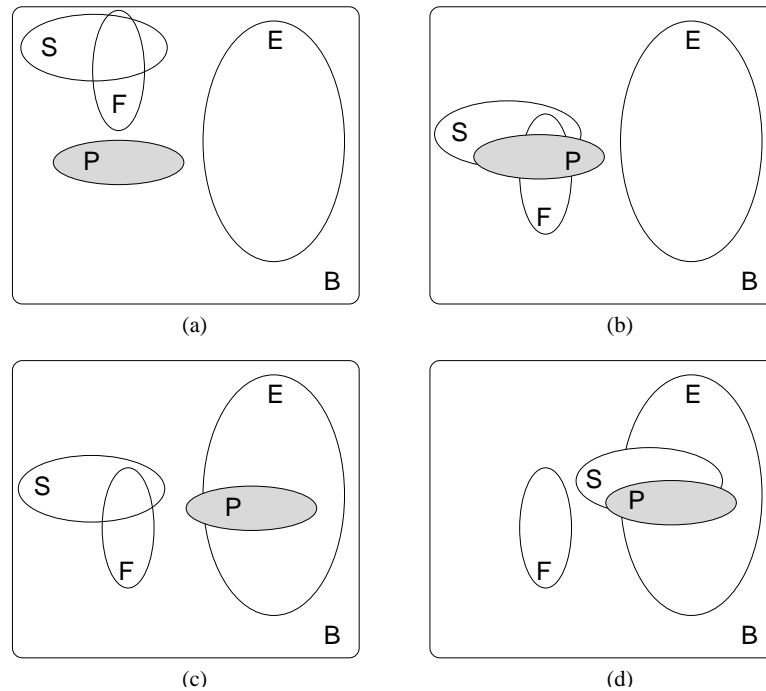
A etapa de monitoração identifica uma situação onde a organização corrente não satisfaz mais as necessidades do SMA: a organização não cumpre sua função de colaborar para a sua finalidade. Em outras palavras, a organização corrente restringe o comportamento dos agentes àqueles que não conduzem à finalidade do SMA (na figura 5.1 são ilustradas algumas das situações onde isso ocorre). Tal situação pode ocorrer quando, por exemplo, o ambiente muda, a finalidade do SMA muda, os requisitos de eficiência não estão sendo satisfeitos, os agentes não estão conseguindo cumprir seus papéis, etc.

O principal problema nesta etapa é identificar que a finalidade não está sendo alcançada porque a organização corrente não o permite. Além da organização, muitas outras razões podem tornar a finalidade insatisfatível (por exemplo, a finalidade do sistema é impossível de ser alcançada, $P = \emptyset$). Na figura 5.1, a situação (c) caracteriza mais precisamente um *falha organizacional*:

$$(P \cap E \neq \emptyset) \wedge (S \cap F \cap P \cap E = \emptyset) \quad (5.1)$$

Mesmo no caso onde se sabe que o problema pode ser solucionado por uma reorganização, uma nova questão surge: que parte da organização está causando o problema? Que nível (EnO, EE, EF, ED) de alteração deve ser adotado?

Em (STRUGEON, 1995), uma situação de reorganização acontece quando uma nova tarefa é apresentada ao sistema e a organização corrente não é adequada para este novo tipo de tarefa. Para determinar que tipo de organização é mais adequado para cada tipo de tarefa, os seguintes atributos da tarefa permitem associá-la diretamente ao melhor tipo de organização: necessidade de conhecimento



Na situação (a) os comportamentos que conduzem à finalidade (P) não são permitidos nem pelo ambiente (E) nem pela organização (representada aqui pelo conjunto $S \cap F$). Contudo, mudar a organização não ajuda muito na solução do problema já que, no máximo, se passará para a situação (b) — não é considerado aqui a possibilidade de alterar o ambiente e, conseqüentemente, o conjunto E . A situação (a), portanto, não caracteriza uma falha organizacional. Em (b), os comportamentos P são permitidos pela organização, mas o ambiente não os permite. Novamente, alterar a organização não resolve o problema. Em (c), é possível realizar os comportamentos P no ambiente corrente, mas a organização não os permite. Neste caso tem-se uma falha organizacional e um processo de reorganização é útil. Em (d), a finalidade poderia ser alcançada na configuração atual ($S \cap E$), mas a especificação funcional não colabora. Novamente, uma reorganização no nível EF pode resolver o problema.

Figura 5.1: Caracterização de alguns tipos de falha organizacional.

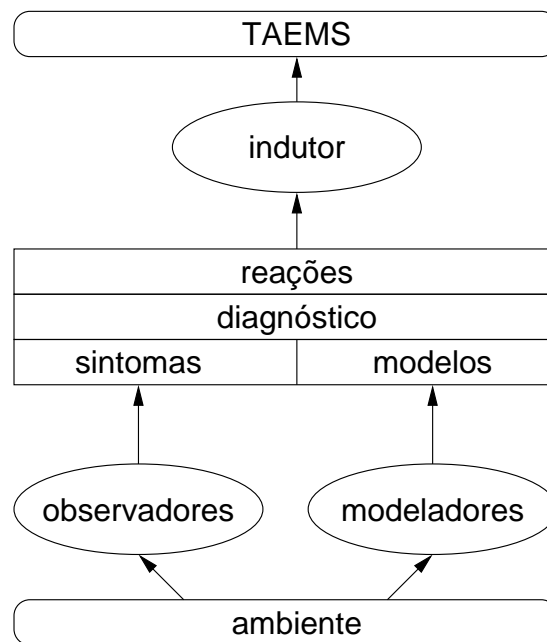
mútuo entre os agentes, de qualidade na execução das tarefas e da necessidade de coordenação. Por exemplo, para uma tarefa onde há maior necessidade de coordenação, uma organização hierárquica pode ser mais apropriada. Em (HURLING; BENYO; LESSER, 2001), um sistema de diagnóstico de execução de tarefas detecta uma situação de falha com base em observações e modelos (figura 5.2). As observações provêm dos agentes e das informações coletadas por eles (por exemplo, excedeu-se o tempo de duração esperado para uma ação) e causam sintomas que serão tratados pelo sistema de diagnóstico. Os modelos (de interação e de uso de recursos, por exemplo) são aprendidos pelo sistema a partir do seu funcionamento e, quando as previsões do modelo aprendido não se confirmam, um sintoma é gerado. Interessante nesta solução é que o sistema de diagnóstico pode ser utilizado por qualquer sistema que tenha sua organização representada por TÆMS sem a necessidade de incluir novos atributos às tarefas (como ocorre, por exemplo, em (STRUGEON, 1995)). Em (SO; DURFEE, 1993), a situação de reorganização (mudança no fator de ramificação da árvore de decomposição da tarefa corrente) é identificada pelas propriedades da tarefa, como por exemplo o seu tamanho. Em (STONE; VELOSO, 1999), algum jogador pode perceber que há uma boa oportunidade para um tipo de formação e decidir adotá-la (reorganização emergente). Como já foi dito, o trabalho de Stone e Veloso também considera o reorganização pré-definida onde o monitoramento é desnecessário.

5.3 Projeto

Uma vez que uma modificação organizacional seja necessária, a etapa de projeto procura construir um conjunto de alternativas para a organização corrente. O projeto deste conjunto de alternativas pode (i) se basear em um biblioteca de organizações pré-definidas ou (ii) criar novas organizações.

No primeiro caso, o problema é identificar qual das organizações pré-definidas é mais apropriada para o problema identificado na monitoração. Em (STRUGEON, 1995), por exemplo, esta decisão é baseada nas características da nova tarefa. Em (STONE; VELOSO, 1999), as características do ambiente (o oponente) ajudam na escolha de uma formação pré-definida.

No segundo caso, as novas EO são criadas, por exemplo, por ferramentas de



Nesta proposta, a monitoração é feita por um sistema de diagnóstico alimentado por sintomas e modelos, o projeto é feito pelo sistema de reação e a seleção pelo indutor que, por fim, altera a organização (descrita em TÆMS).

Figura 5.2: Arquitetura de reorganização proposta por Horling, Benyo e Lesser (2001).

sistemas especialistas (BAZZAN; LESSER; XUAN, 1994; HORLING et al., 2000; HORLING; BENYO; LESSER, 2001), aprendizado (WEISS, 1994), raciocínio baseado em casos (MALONE, 1999), etc.

5.4 Seleção e Implementação

Estas etapas selecionam uma das alternativas apresentadas pela etapa de projeto. O principal problema para a seleção é a definição de critérios de avaliação de propostas, isto é, qual delas terá maior sucesso em solucionar o problema detectado na monitoração. Normalmente, nos trabalhos estudados, essa etapa não é claramente diferenciada da etapa de projeto.

O problema da implantação é como alterar a organização que está em funcionamento. Por exemplo, como um agente irá lidar com o fato que seu papel não existe mais? O que ele fará com os compromissos que assumiu a partir deste papel? Este período de adaptação à nova organização não é considerado por nenhum dos trabalhos de reorganização estudados.

5.5 Conclusões

A partir da concepção de reorganização apresentada e dos trabalhos relacionados, percebe-se que:

- há uma variedade muito grande de problemas reorganizacionais e de abordagens para solucioná-los. Por exemplo, para o proposta de uma nova organização, muitas técnicas distintas são utilizadas (desde busca em um repositório até aprendizado);
- grande parte das soluções apresentadas estão fortemente ligadas ao modelo organizacional adotado. Conseqüentemente, a concepção de reorganização também varia, da simples adoção de papéis até a mudança na descrição das tarefas⁴;

⁴Apesar desta variação, a visão geral do processo de reorganização a partir dos elementos gerais propostos pelo modelo MOISE^+ mostrou-se útil, por exemplo, no caso de definir-se o objeto de mudança. Os conceitos incluídos no MOISE^+ são suficientemente amplos para permitir compreender outros modelos organizacionais a partir destes conceitos.

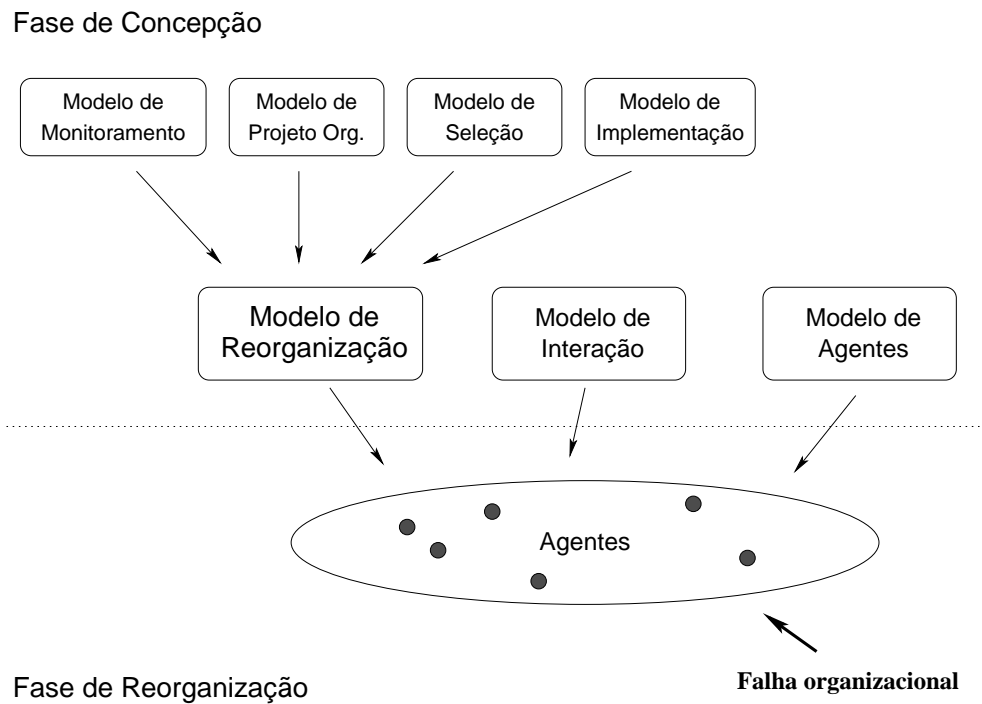


Figura 5.3: Visão geral de um SMA com capacidade de se auto-reorganizar.

- a maioria dos trabalhos considera a forma de reorganização controlada (seja pelos agentes ou pelo usuário);
- poucos trabalhos consideram os problemas de seleção e
- nenhum considera os problemas de implementação.

Uma boa solução para o processo de reorganização de um SMA deveria portanto permitir que várias formas de monitoração, projeto, seleção e implantação pudessem ser utilizadas. Deveria, por exemplo, permitir que tanto a técnica de aprendizado quanto a de sistemas especialistas pudessem ser utilizadas no projeto de novas organizações. Enfim, que a mesma abordagem aberta dos SMA (para cada problema se escolhe os modelos de organização, interação e agentes mais adequados, conforme a figura 1.1, página 4) fosse utilizada no processo de reorganização: para cada necessidade de reorganização os agentes, ou mesmo o projetista do sistema, escolhessem o melhor *modelo de reorganização* (figura 5.3). Comparando a figura 5.3 com a proposta inicial de Alvares e Sichman (1997) (resumida na figura 1.1, página 4), tem-se que a escolha de um modelo de organização é substituída por um modelo de reorganização. Este último modelo

guiaria os agentes na implementação de uma nova organização passando pela escolha de quatro sub-modelos: de monitoração, projeto, seleção e implementação. Deste modo, ao invés do processo de reorganização estar codificado nos agentes (como ocorre na maioria dos trabalhos de reorganização), a reorganização passa a ser explicitamente definida de modo que os próprios agentes podem escolher como realizar sua reorganização, assim como eles escolhem seu modelo de interação.

Dentre os inúmeros tipos de processo de reorganização (pré-definido, controlado ou emergente), objetos de mudança (EnO ou EO), agente de mudança (o(s) agente(s) do sistema ou o usuário) e ferramenta de projeto (sistema especialista, aprendizado, por exemplo), o capítulo seguinte descreve como um processo de reorganização controlado, semelhante ao proposto na figura 5.3, pode ser especificado para sistemas que utilizam o modelo \mathcal{MOISE}^+ como modelo organizacional.

6 Reorganização utilizando o modelo $\mathcal{M}\text{oise}^+$

O capítulo anterior detalhou o processo de reorganização: o que pode mudar em uma organização, quando, como e por quem. A partir das referências estudadas, concluiu-se que há uma grande variedade de problemas e concepções de reorganização. A definição de um mecanismo de reorganização universal que solucione todos os problemas reorganizacionais evidentemente é um trabalho enorme e seria ingenuidade pensar em fazê-lo no escopo de uma tese. Pretende-se portanto avançar na direção de incluir algumas propriedades importantes no processo de reorganização. Particularmente, que um SMA possa escolher uma forma adequada de se reorganizar, ou, pelo menos, possuir um mecanismo de reorganização aberto. Um SMA com mecanismo de reorganização aberto significa que tal sistema permite o uso de várias técnicas de reorganização.

Como o modelo $\mathcal{M}\text{oise}^+$ foi desenvolvido para facilitar a reorganização, este capítulo descreve como um processo de reorganização aberto pode ser concebido em um SMA que tenha o $\mathcal{M}\text{oise}^+$ como modelo organizacional.

6.1 Introdução

Antes de detalhar o mecanismo de reorganização proposto, convém estabelecer seus principais requisitos e propriedades para situá-lo no contexto apresentado no capítulo 5:

- i)* Somente reorganizações no nível de especificação organizacional (EO) serão consideradas. Mudanças no nível da entidade não serão consideradas neste capítulo, pois fazem parte da dinâmica normal de um SMA organizado com o $\mathcal{M}\text{oise}^+$ (conforme visto no capítulo 4).

- ii) A reorganização se dará de forma controlada.
- iii) Dentre as formas controladas, será considerado um processo de reorganização com etapas de monitoração, projeto, seleção e implementação bem definidas.
- iv) Os agentes farão a reorganização (processo endógeno e distribuído).
- v) É assumido que a sociedade (ou parte dela) é organizada segundo o Tipo OC (centrada na organização onde alguns agentes têm capacidade de representá-la). A existência de uma descrição objetiva da organização simplifica o processo de alteração por dispensar tanto o aprendizado de uma descrição subjetiva, quanto a implementação da mudança em todos os agentes. A capacidade de representar e raciocinar sobre tal descrição é uma necessidade para o processo endógeno. Contudo, não se exige que todos os agentes tenham tal capacidade, mas somente os agentes que participam do processo de reorganização.

Como a reorganização é um processo social, como qualquer outro, toda a especificação do processo de reorganização será feita utilizando o \mathcal{MOISE}^+ como ferramenta. Portanto, a notação do \mathcal{MOISE}^+ será utilizada para determinar como um sistema altera sua organização. Seguindo esta abordagem, serão definidos grupos, papéis, missões, planos, obrigações para o conjunto de agentes responsável pela mudança organizacional.

6.2 Estrutura de reorganização

A reorganização é realizada por um grupo criado a partir da especificação ReorgGr , definida na figura 6.1. Os papéis abstratos neste tipo de grupo são: *Monitored*, *Designer* e *Reorg*. *Monitored* é um papel abstrato que será especializado pelos papéis dos agentes que poderão ser monitorados pelo agente *Monitor*. Em outras palavras, todos os agentes que serão monitorados devem ter um papel que especializa *Monitored* e, portanto, estarão sob a autoridade de *Monitor*. O papel de *Designer* também é abstrato e contém propriedades comuns para os dois tipos de projetistas do grupo (*ReorgExpert* e *OrgParticipant*). O papel *Reorg* permite facilmente distinguir o *OrgManager*, descrito na seção 4.1, dos demais

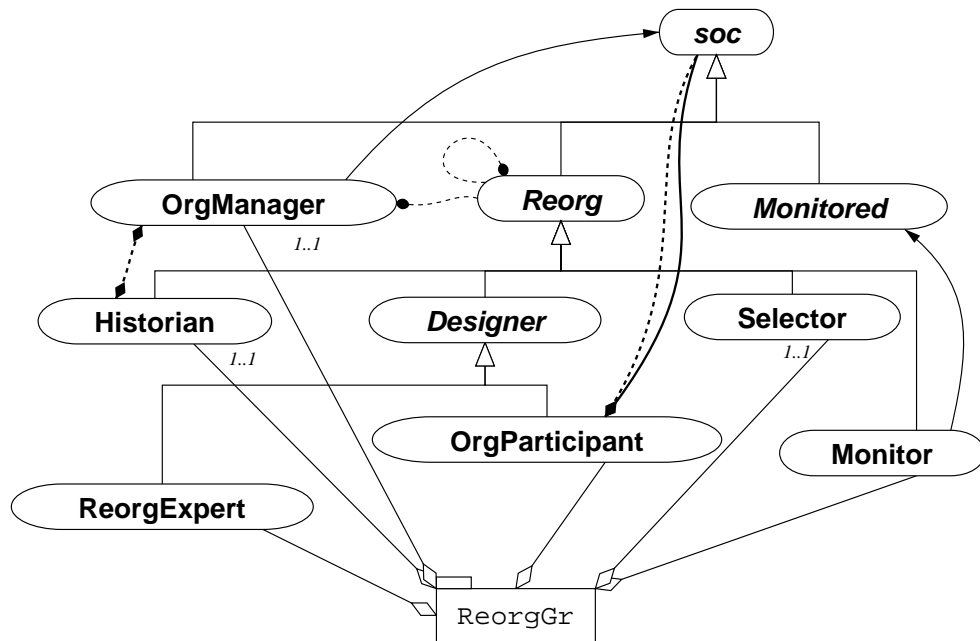


Figura 6.1: Grupo de reorganização.

papéis. Pode-se, por exemplo, determinar que todos os agentes com papel *Reorg* (ou uma especialização dele) podem se comunicar com *OrgManager*. Também por meio do papel *Reorg*, facilmente se estabelece que todos os agentes do grupo podem se comunicar.

A descrição dos papéis não abstratos e sua posição na estrutura do grupo é dada a seguir (a descrição funcional destes papéis é feita nas seções seguintes):

1. **OrgManager**: um, e somente um, agente pode assumir esse papel. Esse agente tem autoridade sobre todos os agentes da sociedade, já que ele tem autoridade sobre *soc*. Esta autoridade é útil para que este agente cumpra com sua principal função: manter o estado corrente da descrição da organização (EO e EnO).
2. **Historian**: o agente que assume este papel mantém a história da organização — um tipo de informação muito útil na monitoração e projeto de novas organizações. Todos os eventos organizacionais descritos na seção 4.2 são registrados. Para ser informado destes eventos, o *Historian* solicita ao *OrgManager* que lhe informe todos os eventos organizacionais ocorridos. Como este papel e *OrgManager* têm atribuições próximas, eventualmente um mesmo agente pode assumir os dois papéis, já que são compatíveis.

3. **Monitor**: os agentes que assumem este papel irão monitorar a organização e identificar situações nas quais uma reorganização deve ser realizada. A ligação (herdada) de comunicação com o *Historian* e a autoridade sobre os agentes *Monitored* são utilizados para esse objetivo.
4. **ReorgExpert**: os agentes com esse papel têm a habilidade (e a obrigação) de analisar a organização corrente, identificar seus problemas e propor alternativas. A princípio, estes agentes não podem participar em outros grupos da sociedade, já que este papel não possui compatibilidades. Os agentes que são convidados a participar do **ReorgGr** como *ReorgExpert* atuam como consultores externos que deveriam ter a habilidade de analisar a sociedade a partir de um ponto de vista global.
5. **OrgParticipant**: como este papel é compatível com *soc*, qualquer agente da sociedade (mesmo fora do grupo **ReorgGr**, pois há uma compatibilidade inter-grupo) pode assumir esse papel. Espera-se que agentes com experiência prática na organização venham a assumir esse papel como um tipo de projetista que possui uma boa visão local, normalmente parcial, da organização.
6. **Selector**: a agente que assume este papel será responsável pela etapa de seleção das propostas de reorganização.

Enfim, nesta proposição, para um SMA poder se reorganizar ele deve ter um grupo **ReorgGr** que irá conduzir o processo de reorganização. Os agentes que formam este grupo são chamados de *configuração de reorganização* pois eles representam as escolhas de modelos de monitoração, projeto, seleção e implantação da reorganização (como sugerido na conclusão do capítulo 5 e na figura 5.3). Neste grupo alguns agentes são de propósito geral (como o *OrgManager* e o *Historian*) e portanto não mudam de uma configuração para outra. Outros agentes são específicos de uma determinada configuração pois têm sua implementação dependente do domínio de aplicação (os monitores, os projetistas e o seletor). Por exemplo, em uma configuração, a monitoração pode ser realizada por detecção de pouco desempenho, o projeto por um sistema especialista e a seleção por votação. Uma outra configuração poderia ser semelhante nos aspectos de monitoração e seleção, mas o projeto seria feito por um agente que utiliza raciocínio baseado em casos.

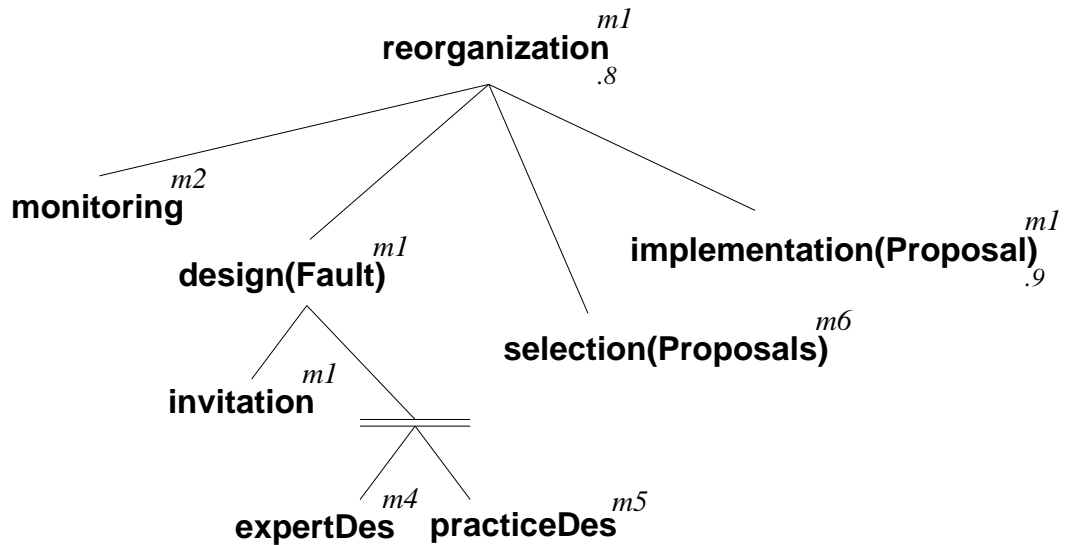


Figura 6.2: Esquema de reorganização.

Tabela 6.1: Relações deônticas para o processo de reorganização

papel	tipo de relação	missão = {metas}
<i>OrgManager</i>	<i>obl</i>	$m_1 = \{reorganization, design, invitation, implementation\}$
<i>Monitor</i>	<i>obl</i>	$m_2 = \{monitoring\}$
<i>ReorgExpert</i>	<i>obl</i>	$m_4 = \{expertDes\}$
<i>OrgParticipant</i>	<i>per</i>	$m_5 = \{practiceDes\}$
<i>Selector</i>	<i>obl</i>	$m_6 = \{selection\}$

6.3 Monitoração

Os agentes que instanciaram o *ReorgGr* irão executar o esquema de reorganização definido na figura 6.2. A ligação deste esquema com o grupo *ReorgGr* é definida pelas relações deônticas da tabela 6.1. Este esquema é controlado pelo *OrgManager* que tem a obrigação de se comprometer com a missão raiz do esquema.

Neste esquema, a etapa de monitoração (missão m_2) é obrigação dos agentes que assumem o papel de *Monitor*. Não será definido aqui como estes agentes irão monitorar a sociedade, pois existem várias maneiras que geralmente são dependentes do domínio de aplicação. Como já comentado na introdução, pretende-se

uma solução onde várias formas de monitoramento possam ser utilizadas, isto é, que vários agentes *Monitor* possam existir e cada um possa, eventualmente, utilizar uma técnica de monitoração distinta.

Apesar de não se especificar como será realizado o monitoramento, o \mathcal{MOISE}^+ e os agentes do grupo $\mathit{ReorgGr}$ oferecem aos monitores várias fontes de informação úteis:

1. a finalidade da sociedade está explicitamente declarada na EnO ;
2. os esquemas sociais são definidos por metas globais que podem ser verificadas;
3. os planos têm uma taxa de sucesso que é atualizada dinamicamente;
4. a boa formação dos grupos pode ser verificada;
5. por meio da definição de papéis como *Historian* e *Monitored* — e o poder que estes papéis implicam — pode-se obter várias outras informações.

Um vez que um agente *Monitor* decide que uma reorganização é necessária, ele informa que a meta *monitoring* está satisfeita. Sendo satisfeita, a meta *design* passa a ser possível e o processo de reorganização avança para a próxima etapa. O *Monitor* também deve enviar uma mensagem para o *OrgManager* descrevendo a falha ocorrida e o nível de reorganização necessário (o que deve mudar na organização corrente). Essa descrição será usada pelo *OrgManager* para instanciar a variável *Fault* da meta *design*.

Exemplo 6.1 Um primeiro exemplo, muito simples, de agente monitor é um agente que periodicamente observa a finalidade da sociedade (γ) e caso ela se altere ele dispara o processo de reorganização. Neste caso, o $\mathit{ReorgGr}$ tem um agente, entre outros, com papel de *Monitor* e que funciona por observação da finalidade. A ligação de comunicação com o *OrgManager* permite consultar a finalidade corrente e lhe enviar a mensagem com a descrição da falha ($\mathit{Fault} = (\mathit{purposeChanged}, \mathit{EO})$, neste exemplo). Este tipo de falha provavelmente irá demandar uma reorganização em toda a especificação organizacional (nível EO).

□

Exemplo 6.2 Um outro agente *Monitor* poderia observar a formação de um grupo específico, como por exemplo o grupo de seleção de candidatos de um programa de pós-graduação (o mesmo definido no exemplo 3.2 da página 52). Este grupo tem um agente *Monitor* incluído, como mostra a figura 6.3. Tal *Monitor* utiliza a seguinte estratégia de monitoração: caso o grupo (gr) fique constantemente mal formado, o *Monitor* percebe que se caracterizou uma situação onde a organização não é adequada e decide disparar o processo de reorganização com $Fault = (\neg wellFormed, grType(gr))$. Tem-se então uma falha com o foco na EE, mais precisamente, na especificação do grupo gr . Para saber quanto tempo e com que periodicidade um grupo fica mal formado, este *Monitor* pode consultar o agente *Historian*.

Nota-se que a definição precisa de “constantemente mal formado” é um parâmetro para este tipo de agente e, apesar deste agente ter um funcionamento útil para vários tipos de aplicação, sua parametrização é dependente de domínio. Assim como também depende do domínio determinar que o fato dos grupos estarem mal formados caracteriza uma falha organizacional. Eventualmente, em algumas aplicações, estar mal formado não caracteriza um problema que justifique uma reorganização.

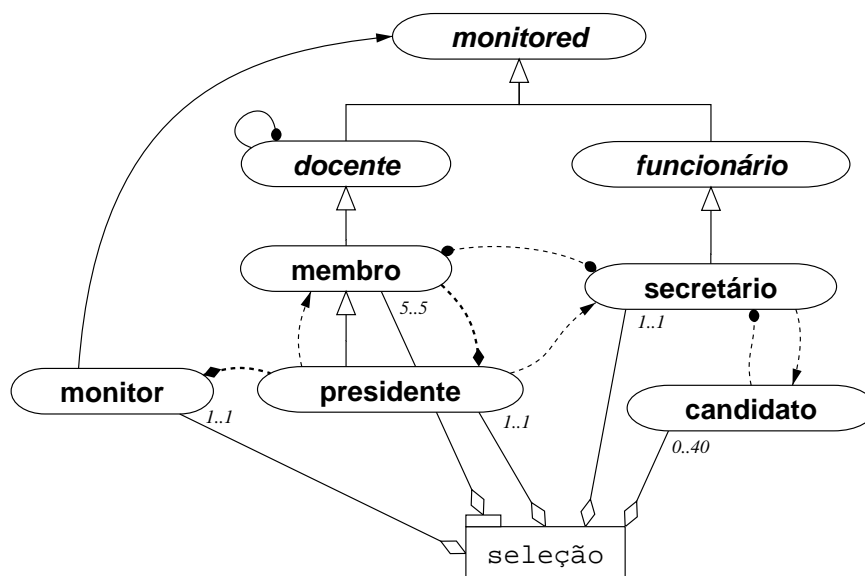
Em sistemas onde este tipo de monitoração é relevante, inclui-se, entre outros, um agente *Monitor* como o apresentado neste exemplo.

□

Exemplo 6.3 Um último exemplo, também muito simples, de agente *Monitor* (incluído em um grupo possivelmente da mesma forma que no exemplo anterior) é um agente que periodicamente observa a taxa de sucesso dos planos de um esquema específico, por exemplo o esquema $sch_{ingresso}$. Quando a taxa de sucesso ficar abaixo de um certo limiar, ele dispara o processo de reorganização com $Fault = (lowSuccess, sch_{ingresso})$.

□

Junto à definição dos Monitores, é necessário especificar a ontologia dos termos utilizados por eles. Por exemplo, é necessário definir o significado de $purposeChanged$, $wellFormed$ e $lowSuccess$. A definição desta ontologia depende da aplicação, ou da configuração de reorganização, e é portanto incluída nos agentes da configuração de reorganização.



O grupo de seleção de candidatos da escola (inicialmente definido na página 54) passou a ter mais um (necessariamente um e no máximo um) agente Monitor que pode ser o mesmo agente que é presidente da comissão de seleção. Os demais papéis passaram a ser especializações de *Monitorado*, dando ao *Monitor* autoridade sobre eles (esta ligação de autoridade está definida no grupo *ReorgGr*, mas foi repetida aqui para facilitar a leitura).

Figura 6.3: Grupo de seleção com um Monitor.

6.4 Projeto

Para satisfazer a meta *design*, o *OrgManager* utiliza o plano definido no esquema de reorganização (sch_{reorg} , figura 6.2). A primeira meta desse plano (*invitation*) é convidar alguns agentes para serem projetistas de uma nova organização. No contexto do *ReorgGr*, aceitar esse convite consiste em aceitar ou o papel de *ReorgExpert* ou o de *OrgParticipant*. O agente que aceitar o papel *ReorgExpert* é obrigado pelas relações deonticas (tabela 6.1) desse papel a se comprometer com a missão m_4 e, conseqüentemente, buscar satisfazer as metas de m_4 (*expertDes* — projeto de organização por experiência em reorganização). Já os agentes que aceitarem o papel de *OrgParticipant* podem (não são obrigados) se comprometer com a missão m_5 . No caso do comprometimento com m_5 , o agente deve satisfazer a meta *practiceDes* (projeto de organização por experiência prática na sociedade).

Como um *OrgParticipant* pode ser também um *Monitor*, há a seguinte ordem de preferência entre as missões destes papéis: $m_2 \prec m_5$. Assim, as atividades de monitoramento (m_2) são socialmente preferíveis às atividades de projeto (m_5) para os agentes que são *OrgParticipant* e *Monitor*.

Como já observado no capítulo 5, o projeto de alternativas à organização corrente pode fazer uso de várias técnicas e ferramentas. No grupo *ReorgGr*, cada uma destas técnicas pode ser implementada como um agente que o *OrgManager* consultará no processo de reorganização. Novamente, pretende-se uma abordagem aberta: tantos projetistas (e técnicas) quanto se julgar necessário podem participar da configuração de reorganização. Como na etapa de monitoração, informações úteis para o projeto podem ser obtidas com o agente *Historian*.

Tal abertura requer que se estabeleça critérios para que um projetista apresente uma *proposta de mudança*:

- i) aceitar o papel de *Designer* e as obrigações que decorrem disso;
- ii) escrever um plano de reorganização que
 - contenha o(s) foco(s) de mudança do plano (nível EO, EE, EF, ED, um grupo, um esquema ou ainda um outro componente da especificação organizacional),

- contenha somente as ações de mudança que serão descritas na seção 6.6,
- altere a organização corrente progressivamente. Por exemplo, não é possível remover toda a EO diretamente, mas primeiro cada papel deve ser removido, depois os grupos e, por fim, a EO, quando vazia, pode ser removida. Portanto, o plano de reorganização deve determinar como a EO corrente será “desmontada”. Entretanto, nem sempre é necessário mudar toda a EO. Pode-se, por exemplo, incluir somente um novo grupo ou mesmo mudar a cardinalidade de um papel.

iii) enviar o plano de reorganização para o *OrgManager* que, com todas as propostas, irá assinalar a meta *design* como satisfeita e instanciar o argumento *Proposals* da meta *selection*.

Exemplo 6.4 No exemplo do grupo de seleção para ingresso na pós-graduação (descrito inicialmente no exemplo 3.2, página 52), um *Monitor*, que monitora a taxa de sucesso dos esquemas (como o monitor do exemplo 6.3), chegou à conclusão de que há um problema no esquema de seleção. Um *Designer* (do tipo *ReorgExpert*), por sua vez, conclui que o problema provém do grupo que executa o esquema e não do esquema. E, segundo seu diagnóstico, o problema é o excesso de comunicação e a falta de autoridade (!). Por fim, ele propõe o seguinte plano de reorganização:¹

- foco na EE, grupo *gtselecao*,

% o funcionário passa a poder se comunicar somente com os membros do grupo de seleção.

Comandos de reorganização:

- remover a ligação de comunicação entre funcionário e docente,
- remover a ligação de comunicação entre funcionário e funcionário,
- adicionar uma ligação de comunicação entre funcionário e membro.

¹O objetivo nesta seção é detalhar como os agentes participam da reorganização, portanto não será detalhado aqui como tal plano de reorganização foi construído.

% O candidato passa a poder se comunicar somente com o secretário (que tem autoridade sobre ele).

Comandos de reorganização:

- remover a ligação de comunicação entre candidato e membro.

% O presidente passa a ter autoridade sobre os membros da comissão.

Comandos de reorganização:

- adicionar uma ligação de autoridade do presidente para membro.

% Algumas cardinalidades são alteradas.

Comandos de reorganização:

- mudar a quantidade de membros na comissão para cinco,
- mudar a quantidade de secretários para um.

Esta proposta de reorganização torna a especificação do grupo de seleção igual a figura 3.4. □

Exemplo 6.5 Para o mesmo problema detectado no grupo de seleção, um outro *Designer*, um *OrgParticipant*, poderia concluir que o fato do esquema de ingresso ter uma taxa de sucesso baixa é a falta de controle sobre a meta g_{14} . Logo, ele decide incluir um plano para que os agentes satisfaçam mais facilmente tal meta. Para isso, a seguinte proposta (bem mais simples que a do exemplo anterior) é submetida:

- foco na EF, grupo $sch_{ingresso}$,

% inclui um plano para a meta g_{14} (o formulário de matrícula preenchido é recebido).

Comandos de reorganização:

- adiciona no esquema o plano $g_{14} = g_{16}, g_{17}, g_{18}$,

- adicionar à missão m_1 a meta g_{16} (o candidato entregou o formulário preenchido para o secretário),
- adicionar à missão m_2 a meta g_{17} (o secretário encaminhou o formulário para o controle acadêmico).

% inclui mais uma missão para um novo secretário (do controle acadêmico)

Comandos de reorganização:

- mudar a quantidade máxima de secretários para três,
 - adicionar a missão m_7 ao esquema,
 - adicionar à missão m_7 a meta g_{18} (o controle acadêmico digitou o pedido de matrícula),
- foco na ED,

Comandos de reorganização:

- adicionar a missão m_7 como obrigação para o papel de secretário.

□

Enfim, nesta seção não foi estabelecido *como* projetar uma alternativa de organização, mas as *condições* para que um agente participe do processo de reorganização e faça propostas de mudança. A forma como estes agentes irão criar suas propostas é dependente da aplicação e da configuração de reorganização.

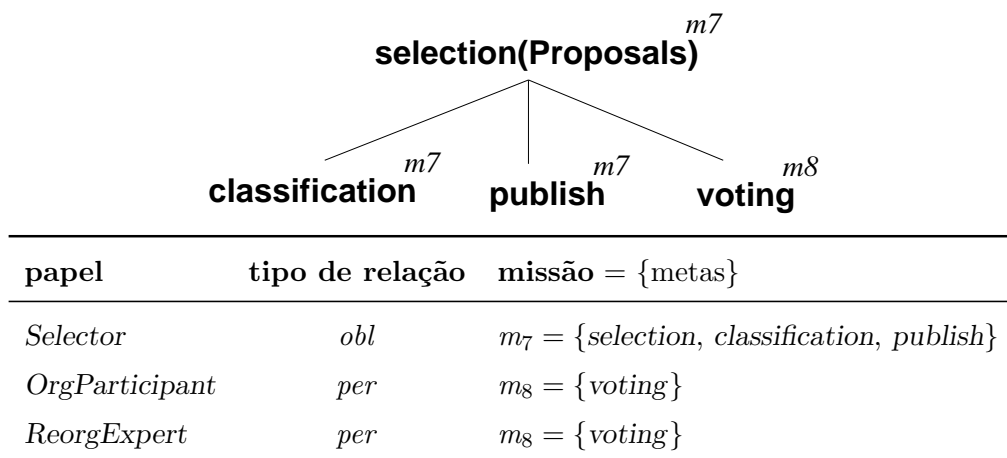
6.5 Seleção

Como nas duas primeiras etapas do processo de reorganização, a forma como a seleção de um plano de mudança ocorre também depende do domínio da aplicação. Apesar disso, nesta seção serão elaboradas duas propostas de seleção que podem ser utilizadas em vários domínios (inclusive nos domínios dos estudos de caso da parte III).

6.5.1 Seleção por votação

Uma possível implementação do agente Selector, responsável por escolher uma das propostas de reorganização apresentadas pelos projetistas, é por meio de votação. Tal proposta consiste de um sistema de votação inicialmente elaborado por Lugo, Hübner e Sichman (2001).

Neste modelo de seleção, um novo esquema, que tem como raiz a meta *selection* do esquema de reorganização, é adicionado à EF:



Este novo esquema tem quatro metas:

- *selection(Proposals)*: esta meta é satisfeita quando uma das proposta de *Proposals* for selecionada para implantação,
- *classification*: esta meta é satisfeita pela classificação prévia das propostas em uma ordem de preferência dada pelo agente *Selector*,
- *publish*: esta meta é satisfeita pela publicação da classificação, e
- *voting*: esta meta é satisfeita quando os agentes comprometidos com a missão m_8 realizarem sua votação para escolher a melhor proposta.

A primeira meta do plano para seleção (*classification*) faz parte da missão do *Selector* (missão m_7). Três critérios são utilizados para classificar as propostas: a experiência de quem fez a proposta, o sucesso das propostas anteriores (habilidade de diagnóstico) e o custo da proposta.

- A experiência de um agente α , considerando o foco da sua proposta p e a versão (indicado por v) da EO, é dada pela função fe :

$$fe(\alpha, p, os_v) = \begin{cases} 0 & \text{se } v = 0 \\ \theta fe(\alpha, p, os_{v-1}) + fre(\alpha, p, os_v) + fme(\alpha, p, os_v) & \text{se } v \neq 0 \end{cases} \quad (6.1)$$

onde:

- α é um agente que participou da EO em alguma das suas versões anteriores a v (de os_1 a os_v),
- θ representa a relevância da participação em versões da EO anteriores a v . Por exemplo, se θ é 1, a experiência em versões passadas da EO tem a mesma importância que na versão atual; se θ é 0, experiências anteriores não tem valor.
- $fre(\alpha, p, os_v)$ é uma função que retorna o número de papéis que o agente α assumiu na EnO quando a EO estava na versão v . Este cálculo considera o foco da proposta p . Se o foco é um único papel, fre conta somente as vezes que o agente assumiu este papel; se o foco é um grupo, a função conta as vezes que o agente assumiu um papel deste grupo; se o foco está na ED, fre conta as vezes que α assumiu o papel da relação deontica; e, finalmente, se o foco é na EF, $fre = 0$.
- $fme(\alpha, p, os_v)$ é uma função que retorna o número de missões que o agente α se comprometeu até a versão v da EO. De forma análoga à função fre , é considerado o foco da proposta p .

Normalmente, a experiência dos *ReorgExperts* é zero pois eles não têm participação na EnO.²

- a habilidade de diagnóstico de um agente α é dada por

$$da(\alpha) = \sum_{os_v \in OS_\alpha} \epsilon(endTime(os_v) - startTime(os_v)) \quad (6.2)$$

onde:

²A exceção seria o caso onde o próprio grupo **ReorgGr** ou o esquema *sch_{reorg}* estivesse sendo revisado.

- α é um agente que já fez alguma proposta e esta proposta foi implementada,
- o conjunto OS_α contém todas as EO que o agente α alterou, isto é, a EO depois da execução de um plano de reorganização proposto por α ,
- ϵ representa a relevância do tempo que uma EO ficou ativa: da sua implementação — *startTime* — até o momento onde foi objeto de um processo de reorganização — *endTime* (contado em dias, por exemplo).

Resumidamente, a habilidade de diagnóstico é calculada sobre o tempo que as EO propostas ficaram ativas.

- O custo da proposta p é dado por:

$$cost(p) = \phi (rolePlayersRemoved(p) + missionPlayersRemoved(p)) \quad (6.3)$$

onde:

- ϕ é a relevância do custo dos planos para o SMA,
- $rolePlayersRemoved(p)$ é o número de pares (agente \times papel) que serão removidos caso a proposta p seja implementada (estes pares são dados pela função *agRole* da EnO).
- $missionPlayersRemoved(p)$ é o número pares (agente \times missão) que serão removidos caso a proposta p seja implementada (estes pares são dados pela função *agMis* da EnO).

Portanto, o custo de uma proposta é basicamente o custo de remover os papéis e missões dos agentes. Incluir um novo papel ou remover um grupo onde não há agentes não tem custo algum segundo esta proposta, pois consiste da simples alteração de uma estrutura de dados sem conseqüências para os agentes e suas atividades.

Finalmente, a classificação das propostas é feita de acordo com o valor da seguinte função:

$$position(p, \alpha, os_v) = fe(\alpha, p, os_v) + da(\alpha) - cost(p) \quad (6.4)$$

Para satisfazer a segunda meta do plano de seleção (a meta *publish*), o *Selector* inclui às propostas classificadas a proposta “no change” e as publica para que os *ReorgExpert* e os *OrgParticipant* comprometidos com a missão m_8 possam votar. Cada agente pode votar em uma proposta e, após a votação, a pontuação final de cada proposta p é dada por

$$votes(p) = \sum_{\alpha \in E(p)} ge(\alpha, os_v) + da(\alpha) + 1 \quad (6.5)$$

onde $E(p)$ é o conjunto dos agentes que votaram na proposta p e ge (experiência geral — sem foco) é dada por

$$ge(\alpha, os_v) = \begin{cases} 0 & \text{se } v = 0 \\ \theta ge(\alpha, os_{v-1}) + \\ re(\alpha, os_v) + me(\alpha, os_v) & \text{se } v \neq 0 \end{cases} \quad (6.6)$$

onde $re(\alpha, os_v)$ retorna o número de papéis que um agente assumiu na versão v da EO, sem considerar o foco; e $me(\alpha, os_v)$ retorna o número de missões executadas nesta mesma versão da EO. Nota-se, portanto, que a experiência e a habilidade de diagnóstico do eleitor aumentam o valor do seu voto.

A proposta com maior valor para $votes(p)$ é selecionada para implementação.

Exemplo 6.6 Suponha-se que na escola existam os agentes da tabela 6.2, que inclui também as missões e papéis que cada agente desempenhou, e o SMA escola (a EnO) teve a história que está resumida na tabela 6.3. Por concisão, neste exemplo somente a ação de poucos agentes serão considerados.

Na etapa de projeto, foram apresentadas duas propostas: a do exemplo 6.4 feita pelo agente E e a do exemplo 6.5 feita pelo agente P . A classificação das propostas, sobre a qual será realizada a votação, é a seguinte:

proposta	foco	fe	da	$cost$	$position$
$E.p1$	$gr_{selecao}$	$0,5*0+0+0=0$	$0,5*5=2,5$	$2*(1+1)=4$	-1,5
$P.p2$	$sch_{ingresso}$	$0,5*1+0+1=1,5$	0	$2*(0+0)=0$	1,5
no change		0	0	0	0

Os parâmetros utilizados foram: $\theta = 0,5$ (a participação em versões anteriores tem metade do valor), $\epsilon = 0,5$ (cada dia que uma EO ficou ativa conta 0,5) e

Tabela 6.2: Papéis e missões assumidos pelos agentes do exemplo escola

agente	papéis	grupo	missões	EO
<i>E</i>	<i>ReorgExpert</i>	$gr0_{reorgGr}$	$sch_{reorg}.m_2$	os_1
	<i>ReorgExpert</i>	$gr1_{reorgGr}$	$sch_{reorg}.m_2$	os_3
<i>P</i>	<i>OrgParticipant</i>	$gr1_{reorgGr}$	$sch_{reorg}.m_5$	os_3
	<i>Candidato</i>	$gr2_{selecao}$	$sch_{ingresso}.m_1$	os_2
	<i>Candidato</i>	$gr3_{selecao}$	$sch_{ingresso}.m_1$	os_3
<i>A</i>	<i>Secretário</i>	$gr2_{selecao}$	$sch_{ingresso}.m_2$ e $sch_{ingresso}.m_3$	os_2
	<i>Secretário</i>	$gr3_{selecao}$	$sch_{ingresso}.m_2$	os_3
<i>B</i>	<i>Secretário</i>	$gr3_{selecao}$	$sch_{ingresso}.m_3$	os_3
<i>C</i>	<i>Presidente</i>	$gr3_{selecao}$	$sch_{ingresso}.m_4$ e $sch_{ingresso}.m_5$	os_3
	<i>OrgParticipant</i>	$gr1_{reorgGr}$	$sch_{reorg}.m_6$	os_3
	<i>Membro</i>	$gr3_{selecao}$	$sch_{ingresso}.m_5$	os_3
<i>D</i>	<i>Membro</i>	$gr3_{selecao}$	$sch_{ingresso}.m_5$	os_3
	<i>OrgParticipant</i>	$gr1_{reorgGr}$	$sch_{reorg}.m_6$	os_3

Tabela 6.3: Resumo da história da entidade organizacional escola

dia	evento	EO
1	a escola foi criada com os_1	os_1
3	uma reorganização, gerenciada pelo $gr0_{reorgGr}$, mudou a EO para os_2 , tal modificação deu-se a partir de uma proposta do agente <i>E</i>	os_2
4	o grupo de seleção $gr2_{selecao}$ foi criado. Somente o agente <i>A</i> participou deste grupo como secretário com duas missões	os_2
7	o grupo $gr2_{selecao}$ foi extinto	os_2
8	uma reorganização, gerenciada pelo $gr0_{reorgGr}$, mudou a EO para os_3	os_3
9	foi criado o grupo $gr3_{selecao}$	os_3
10	começou uma nova reorganização de os_3 , gerenciada pelo grupo $gr1_{reorgGr}$. Neste grupo estão os agentes <i>C</i> , <i>D</i> , <i>E</i> e <i>P</i> , com as missões indicadas na tabela 6.2	os_3

$\phi = 2$ (relevância do custo).

O cálculo da tabela tem as seguintes explicações:

- $fe(E, p1, os_3) = 0$ porque o agente E não tem experiência na organização (ele é um consultor externo).
- $fe(P, p2, os_3) = 1,5$ porque
 - $\theta fe(P, p2, os_2) = 0,5$ o agente P participou na versão anterior (2) da EO como candidato no grupo $gr_{selecao}^2$ com a missão $sch_{ingresso}.m_1$. Como o foco do plano $p2$ é um esquema, o fato de assumir um papel não é considerado. É considerado somente o comprometimento com missões do esquema $sch_{ingresso}$ (o foco da proposta $p2$).
 - $fre(P, p2, os_3) = 0$ porque o foco da proposta é na EF.
 - $fme(P, p2, os_3) = 1$ porque o agente está assumindo uma missão na versão corrente do esquema que ele propôs mudar.
- $da(E) = \sum_{os_v \in \{os_2\}} \epsilon(endTime(os_v) - startTime(os_v))$
 $= 0,5 * (endTime(os_2) - startTime(os_2))$
 $= 0,5 * (8 - 3)$
 $= 2,5$
 o agente E teve uma proposta implementada por cinco dias (do dia 3 ao dia 8).
- $da(P) = 0$ pois o agente P nunca teve uma proposta implementada.
- $cost(p1) = 2 * (rolePlayersRemoved(p1) + missionPlayersRemoved(p1))$
 $= 2 * (1 + 1)$
 $= 4$
 a implementação da proposta $p1$ implica que um agente (um dos secretários) perca seu papel e sua missão, já que $p1$ permite apenas um secretário e, no momento, há dois.
- $cost(p2) = 2 * (rolePlayersRemoved(p1) + missionPlayersRemoved(p1))$
 $= 2 * (0 + 0)$
 $= 0$
 a implementação da proposta $p2$ não causa perda de papéis ou missões, esta proposta somente acrescenta um novo plano ao esquema de ingresso.

Os votos (e seus pesos) foram os seguintes

proposta	eleitor	ge do eleitor	da	valor do voto ($ge+da+1$)
<i>E.p1</i>	<i>E</i>	0	2,5	3,50
	<i>D</i>	$0,5*0+1+1$	0	3,00
	<i>total</i>			<i>6,50</i>
<i>P.p2</i>	<i>P</i>	$0,5*2+1+1$	0	4,00
	<i>C</i>	$0,5*0+2+3$	0	6,00
	<i>total</i>			<i>10,00</i>
“no change”	<i>total</i>			<i>0,00</i>

Observa-se que somente os agentes do grupo $gr1_{reorgGr}$ podem votar pois este é o grupo que está conduzindo a reorganização. A participação nos papéis e missões do grupo de reorganização não contam como experiência.

Considerada esta classificação, a proposta do agente *P* será implementada.

□

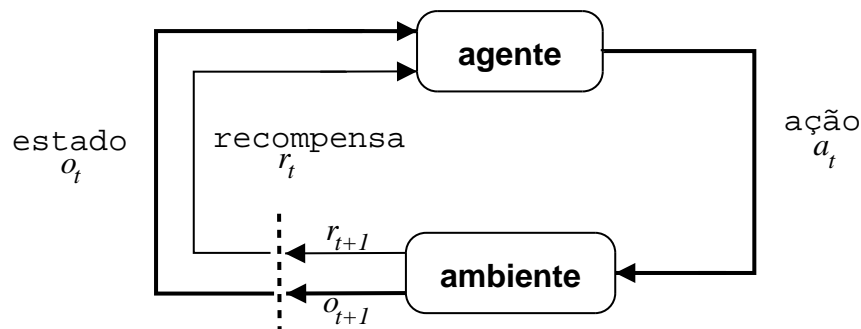
6.5.2 Seleção por aprendizado

Uma segunda proposta de implementação para o agente *Selector* é a utilização de técnicas de simulação e aprendizado. A partir de simulações do domínio de aplicação e de suas reorganizações, o *Selector* vai aprendendo a escolher as melhores propostas de reorganização.

Considerando um domínio onde as decisões do *Selector* podem ser avaliadas, a técnica de *aprendizado por reforço*³ pode ser utilizada para que este agente aprenda a tomar as melhores decisões.⁴ Nesta técnica, o processo de decisão do agente é visto, resumidamente, da seguinte forma (ilustrado na figura 6.4): a cada reorganização, o *Selector* percebe o ambiente (a organização corrente), executa uma ação (a escolha de uma das propostas de reorganização enviadas pelos projetistas), tal ação altera o ambiente (a proposta escolhida é implementada), a decisão (a nova organização) estabelece uma recompensa, ou punição, para o

³As seguintes referências foram utilizadas no desenvolvimento desta seção: (KAELBLING; LITTMAN; MOORE, 1996; SUTTON; BARTO, 1998; MITCHELL, 1997).

⁴Ao contrário do aprendizado supervisionado, no aprendizado por reforço não há necessidade de uma oráculo que diga qual seria a decisão correta. No lugar da decisão ideal, o agente recebe apenas uma recompensa por sua decisão.



O agente percebe o ambiente o no momento t , toma a decisão de fazer a , que altera o ambiente. O ambiente passa para o estado o_{t+1} e dá uma recompensa r_{t+1} para este novo estado (SUTTON; BARTO, 1998).

Figura 6.4: Agente no contexto de aprendizado por reforço.

Selector (se a nova organização melhorou ou não o desempenho do sistema). O objetivo do agente *Selector* é, a partir das recompensas que recebe, ir melhorando seu processo de decisão procurando maximizar a soma das recompensas acumuladas no decorrer do processo. No início, quando o agente ainda não fez muitas decisões, a única opção é a tentativa e erro. À medida que vai ganhando experiência no processo de decisão, a “tentativa e erro” passa, gradativamente, a dar lugar às opções com maior chance de recompensa.

6.5.2.1 Processo de Decisão de Markov

A definição mais precisa do processo de aprendizado pode ser feita modelando-o como um Processo de Decisão de Markov (MDP). Um MDP consiste dos seguintes elementos:

- um conjunto de estados do ambiente O ,
- um conjunto de ações A ,
- uma função que determina a recompensa imediata para as decisões do agente ($r : O \times A \rightarrow \mathbb{R}$, onde \mathbb{R} é o conjunto dos números reais), $r(o, a)$ é a recompensa por escolher a ação a no estado o ,
- uma função que representa as transições de estado do ambiente ($t : O \times A \rightarrow \Pi(O)$), onde um elemento de $\Pi(O)$ é a distribuição de probabilidades sobre

o conjunto O (isto é, $\Pi(O) =_{\text{def}} O \rightarrow [0, 1]$). A função $t(o, a, o')$ representa a probabilidade do ambiente mudar do estado o para o estado o' caso a ação a seja executada.

Como a função de transição sugere, em um MDP, o próximo estado do ambiente é determinado somente pelo estado corrente e pela ação escolhida e os estados ou ações anteriores não são necessários na função de transição. A função também indica que o ambiente é não determinístico (a mesma ação pode levar a estados diferentes em ocasiões diferentes) e é estacionário (a probabilidade de uma ação levar a um estado específico não muda).

Esta modelagem permite definir a valor máximo de recompensas (incluindo as recompensas futuras) para um determinado estado o do ambiente como⁵:

$$V^*(o) = \max_a \left(r(o, a) + \gamma \sum_{o' \in O} t(o, a, o') V^*(o') \right) \quad (6.7)$$

onde o melhor valor para o estado o é dado pela escolha da ação a que maximiza o valor da recompensa imediata r mais a soma, descontada por γ , das recompensas futuras conseguidas a partir dos estados possíveis o' depois da execução da ação a . O parâmetro γ ($0 \leq \gamma \leq 1$) determina um desconto para os valores futuros (se $\gamma = 0$ recompensas futuras não têm valor, se $\gamma = 1$ recompensas futuras têm o mesmo valor que a recompensa imediata).

Para o agente que toma as decisões, existem várias *políticas* possíveis. Da mesma forma que a função t representa o ambiente da figura 6.4, uma política π representa o comportamento do agente da figura através do mapeamento entre os estados do mundo O e as ações A ($\pi : O \rightarrow A$). Dentre as várias políticas possíveis, interessa ao agente encontrar a política ótima (denotada por π^*). No caso de MDPs onde o modelo de transição t é conhecido, esta política é dada por:

$$\pi^*(o) = \arg \max_a \left(r(o, a) + \gamma \sum_{o' \in O} t(o, a, o') V^*(o') \right) \quad (6.8)$$

Técnicas de programação dinâmica podem ser utilizadas para encontrar V^* e π^* .

⁵Esta forma de cálculo é chamada de “modelo de horizonte infinito com desconto” (Kaelbling; Littman; Moore, 1996).

6.5.2.2 Q-Learning

Em muitos domínios, não se conhece *a priori* como ocorrem as transições do ambiente (a função t), o que inviabiliza o uso de ferramentas de programação dinâmica. O algoritmo Q-Learning permite encontrar uma política ótima mesmo desconhecendo-se as transições do ambiente (SUTTON; BARTO, 1998).

No Q-Learning, ao invés de considerar o valor de um estado ($V(o)$), passa-se a considerar o valor de escolher uma ação em um estado: o valor $Q(o, a)$. Considerando que $Q^*(o, a)$ denota o valor escolher a ação a no estado o e depois continuar escolhendo as ações ótimas, tem-se que

$$Q^*(o, a) = r(o, a) + \gamma \left(\sum_{o' \in O} t(o, a, o') \max_{a'} Q^*(o', a') \right) \quad (6.9)$$

os valores de V^* e π^* podem ser redefinidos da seguinte forma:

$$V^*(o) = \max_a Q^*(o, a) \quad (6.10)$$

$$\pi^*(o) = \arg \max_a Q^*(o, a) \quad (6.11)$$

O algoritmo da figura 6.5 aproxima o valor Q^* a partir das experiências do agente, que tem a função de substituir a função de transição (no lugar da função t foi colocada a transição observada no ambiente, compare a linha 10 do algoritmo e a equação (6.9)). Entre as propriedades do algoritmo Q-Learning, as seguintes são as mais relevantes:

1. se cada ação for escolhida um número infinito de vezes em cada estado e o valor de α decair gradativamente, o valor de Q converge para Q^* ,⁶
2. a política que o agente utiliza para escolher as ações (linha 7 do algoritmo) não interfere no aprendizado de Q^* (desde que a condição do item anterior seja mantida).⁷ Poderia ser, por exemplo, escolher uma ação aleatoria-

⁶Normalmente se pode relaxar um pouco na condição de infinitas execuções de a à medida que se consegue uma política aceitável (não necessariamente ótima). Outra dificuldade gerada por esta propriedade é determinar o decaimento de valor de α pois não existe uma resposta geral para este problema. Normalmente este problema é solucionado empiricamente.

⁷Algoritmos de aprendizagem por reforço com esta característica são chamados de “off-policy”.

```

1: inicializar os valor de  $Q(o, a)$  arbitrariamente
2:                               % Q pode ser implementado como um matriz onde uma dimensão
3:                               % é dada pelos valores de  $o$  e a outra pelos valores de  $a$ 
4: for all episódios do           % episódio são as transições entre um estado inicial e final
5:   observar o estado inicial  $o$ 
6:   repeat
7:     escolher uma ação  $a$  para o estado  $o$ 
8:     executar a ação  $a$ 
9:     observar a recompensa  $r$  pela decisão e o novo estado  $o'$ 
10:     $Q(o, a) \leftarrow Q(o, a) + \alpha (r(o, a) + \gamma \max_{a'} Q(o', a') - Q(o, a))$ 
11:                                     %  $\alpha$  é a taxa de aprendizado
12:     $o \leftarrow o'$ 
13:   until  $o$  ser um estado final
14: end for

```

Figura 6.5: Algoritmo Q-Learning

mente. Porém, normalmente se utiliza uma política que inicialmente escolhe aleatoriamente as ações e, à medida que vai aprendendo, passa a utilizar cada vez mais as decisões determinadas pela política derivada de Q . Esta estratégia inicia portanto *explorando* (tentar uma ação mesmo que ela não tenha o maior valor de Q) e termina *explotando* (escolher a ação que tem o maior valor de Q)

Por meio de simulação do ambiente (geração dos episódios) e do algoritmo de Q-Learning, o agente *Selector* pode conseguir uma função Q próxima de Q^* e que pode ser utilizada para tomar decisões ótimas (π^* da equação (6.11)). No caso da seleção de propostas de reorganização, o MDP é formado por:

- o conjunto de estados O é dependente da aplicação;
- o conjunto de ações A pode ser formado ou
 - a) pelas propostas de reorganização (o valor Q seria associado à proposta) ou
 - b) pelos nomes dos agentes que fizeram as propostas (o valor Q seria associado ao mentor da proposta);
- a função de recompensa r depende da aplicação; e
- a função de transição é dispensada pelo uso do Q-Learning.

6.5.3 Comparação entre as duas propostas de seleção

As duas propostas de desenvolvimento do agente *Selector* que foram apresentadas não apresentam diferenças muito significativas. As maiores diferenças recaem na sua implementação.

Ambas permitem ao SMA explorar novas organizações mesmo que não se tenha evidências de que essa nova organização irá trazer uma recompensa maior.⁸ No caso do Q-Learning, fica evidente a possibilidade de exploração. No caso da votação, a exploração é possível pela inclusão de eleitores que eventualmente votam em uma proposta com baixa classificação.

Ambas permitem avaliar o propositor das alterações organizacionais. No Q-Learning, basta considerar o conjunto de ações como sendo composto pelos agentes propositores. No caso da votação, o peso dos votos é dado por uma avaliação da *reputação* do propositor (se fez boas propostas, se participa, etc.). Porém, a votação considera apenas a reputação na classificação das propostas. A proposta em si, o *conteúdo*, não é explicitamente considerada. Claro que os eleitores podem analisar o conteúdo das propostas antes de votar, mas isso está implícito no processo de votação. Por outro lado, o Q-Learning permite considerar explicitamente a avaliação do conteúdo das propostas bastando considerar o conjunto de ações como sendo formado pelas propostas possíveis.⁹

Ambas têm um modelo de recompensa. Na votação, a recompensa está representada no tempo que um proposta ficou ativa, isto é, quanto tempo ela persistiu antes de ser alvo de uma monitoração. No Q-Learning, também há uma recompensa, expressa pela função r , mas este valor depende do domínio.

Resumidamente, o Q-Learning tem as seguintes vantagens:

- a estratégia de exploração está explícita,
- é possível considerar tanto as propostas quanto os propositores na seleção das propostas e
- a recompensa pode considerar aspectos do domínio de aplicação,

⁸Esta possibilidade não existe em vários trabalhos de reorganização, onde, conforme visto no capítulo 5, há interesse apenas na exploração.

⁹Esta opção traz um problema de implementação, já que o conjunto de propositores é finito e o conjunto de proposta normalmente é infinito.

e a seguinte desvantagem:

- há necessidade de um simulador do ambiente.¹⁰

6.6 Implementação

Depois do agente *Selector* terminar a etapa de seleção de uma proposta de mudança (utilizando eventualmente uma das duas técnicas sugeridas — votação ou aprendizado), a próxima etapa do processo de reorganização é a implementação da proposta selecionada. A implementação de um plano de reorganização p consiste da execução do plano p pelo *OrgManager*. Após a execução do plano p sobre a versão corrente da EO os_v , a EO terá a versão os_{v+1} . Esta história da EO também é chamada de *dinâmica da especificação organizacional* que, no caso da proposta apresentada neste capítulo, é gerenciada pelo grupo de reorganização. Um SMA tem portanto

- a dinâmica da entidade organizacional (já visto na seção 4.2) e
- a dinâmica da especificação organizacional.

Assim como as ações sobre a EnO produzem eventos organizacionais, as ações sobre a EO produzem *eventos reorganizacionais*. Não convém detalhar aqui todas estas ações. Portanto, como exemplo, somente aqueles eventos relacionados à noção de papel serão descritos em detalhe a seguir.

Evento	adição de uma definição de papel <code>addRoleDefinition(roleId)</code>
Argumentos	ρ : a identificação do novo papel (roleId)
Pré-condições	
Efeitos	$\mathcal{R}_{ss}^{t+1} = \mathcal{R}_{ss} \cup \{\rho\}$

¹⁰Poderia-se aprender sem o simulador, utilizando as experiências reais para alimentar o Q-Learning. Considerando que mudanças organizacionais não são muito frequentes e que podem causar várias mudanças no sistema, o aprendizado, em geral, é muito lento tornando o desempenho do sistema muito ruim por um longo período de tempo.

O novo papel é incluído no conjunto de papéis da especificação organizacional \mathcal{R}_{ss} sem qualquer pré-condição. Ou seja, o conjunto de papéis da EE (\mathcal{R}_{ss}) após o evento (em $t + 1$) será o conjunto \mathcal{R}_{ss} antes do evento (em t) mais o papel ρ .

Evento	adicionar herança a um papel $\text{addSuperRole}(\text{roleId}, \text{superRoleId})$
Argumentos	ρ_e : a identificação sub-papel (roleId) ρ_s : a identificação do super-papel (superRoleId)
Pré-condições	$\neg (\rho_e \sqsubset \rho_s)$ (1) $\rho_e \in \mathcal{R}_{ss}$ (2) $\rho_s \in \mathcal{R}_{ss}$ (3)
Efeitos	$\sqsubset^{t+1} = \sqsubset \cup \{(\rho_s, \rho_e)\}$

Para um papel especializar outro, este último não pode especializar o primeiro (a relação de herança é anti-simétrica) e os dois papéis devem existir previamente ao evento.

Evento	adicionar um papel a um grupo $\text{addRole}(\text{grSpecId}, \text{roleId})$
Argumentos	gt : a identificação do grupo (grSpecId) ρ : a identificação do papel (roleId)
Pré-condições	$\rho \in \mathcal{R}_{ss}$
Efeitos	$gt.\mathcal{R}^{t+1} = gt.\mathcal{R} \cup \{\rho\}$ $gt.np^{t+1} = gt.np \cup \{\rho \mapsto (0, \infty)\}$

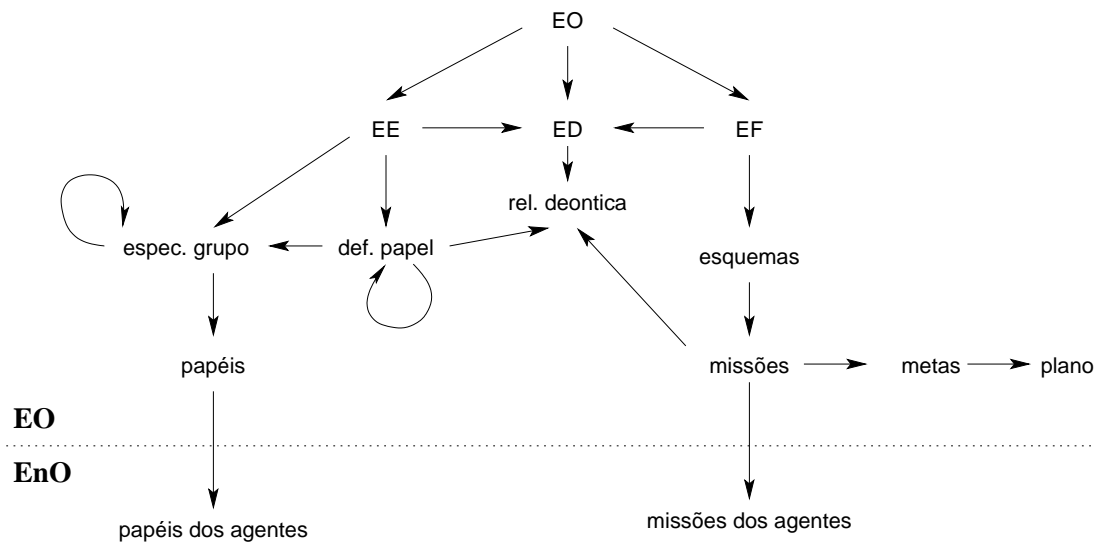
O papel ρ passa a pertencer ao conjunto de papéis possíveis no grupo gt e tem, nesse grupo, a cardinalidade *default*. Como consequência desse evento, o papel ρ deixa de ser abstrato.

Evento	remover uma definição de papel removeRoleDefinition(roleId)
Argumentos	ρ : a identificação do papel a ser removido(roleId)
Pré-condições	$\rho \neq \rho_{soc}$ (1) $\forall gt, m, tc \bullet$ $\neg \exists \rho \in \mathcal{R}_{gt}$ (2) $\neg \exists \rho \sqsubset \rho_e$ (3) $\neg \exists obl(\rho, m, tc) \in \mathcal{DS}$ (4) $\neg \exists per(\rho, m, tc) \in \mathcal{DS}$ (5)
Efeitos	$\mathcal{R}_{ss}^{t+1} = \mathcal{R}_{ss} - \{\rho\}$

Para remover um papel, as seguintes pré-condições devem ser satisfeitas por ele:

- não pode ser o papel social (ρ_{soc}) (1),
- não deve fazer parte de nenhum grupo (2),
- não deve ter especializações (3), e
- não deve ter relações deônticas ((4) e (5)).

Portanto, para remover um papel, primeiro deve-se removê-lo dos grupos, depois remover suas especializações e ligações, para então poder remover seu identificador do conjunto de papéis (veja figura 6.6).



Para remover uma EO, a EE, ED e EF devem estar vazias. Para remover uma EE, suas especificações de grupo e de papéis devem ter sido removidas. Para remover uma definição de papel deve-se remover primeiro suas especializações, suas relações deonticas e sua ocorrência nos grupos. Para remover uma especificação de grupo, deve-se remover antes seus sub-grupos e seus papéis. Para remover um papel da EO, deve-se remover esse papel dos agentes (na EnO) conforme descreve a figura 4.2.

Figura 6.6: Dependência para remoção dos componentes da EO.

Evento	remover um papel de um grupo $removeRole(grSpecId, roleId)$
Argumentos	gt : a identificação do grupo ($grSpecId$) ρ : a identificação do papel ($roleId$)
Pré-condições	$\rho \in \mathcal{R}_{gt}$ (1) $\forall gr \in \{g \mid grType(g) = gt\} \bullet grPlayers(gr, \rho) = \emptyset$ (2)
Efeitos	$\mathcal{R}_{gt}^{t+1} = \mathcal{R}_{gt} - \{\rho\}$ $np_{gt}^{t+1} = np_{gt} - \{\rho \mapsto np_{gt}(\rho)\}$ $\mathcal{L}_{gt}^{t+1} = removeRoleLinks(\rho, \mathcal{L}_{gt})$ $\mathcal{C}_{gt}^{t+1} = removeRoleCompat(\rho, \mathcal{C}_{gt})$

Para remover um papel ρ de um grupo gt , o papel deve pertencer ao grupo (1) e, nos grupos instanciados a partir de gt , nenhum agente deve ter assumido o papel ρ (2). Como consequência desse evento, o papel deixa de existir bem como suas ligações, compatibilidades e cardinalidade. Aqui surge um fato importante: para mudar a EO é necessário, eventualmente, mudar a EnO. Assim, para remover um papel de uma especificação de grupo, eventualmente, alguns agentes devem perder esse papel nas instâncias desta especificação — um mudança na EnO.

6.7 Conclusões

A figura 6.7 resume a proposta de reorganização apresentada, que é constituída dos modelos enumerados a seguir.

- Como modelo de monitoração, existe um conjunto de agentes responsáveis e com autoridade para iniciar uma reorganização. Várias técnicas de monitoração podem portanto ser utilizadas através dos vários agentes de monitoração. O princípio é ter um processo de reorganização **aberto** onde a configuração de reorganização pode incluir, durante o funcionamento do sistema, novos agentes.
- Como modelo de projeto de novas organização, tem-se o mesmo princípio: vários agentes/técnicas.

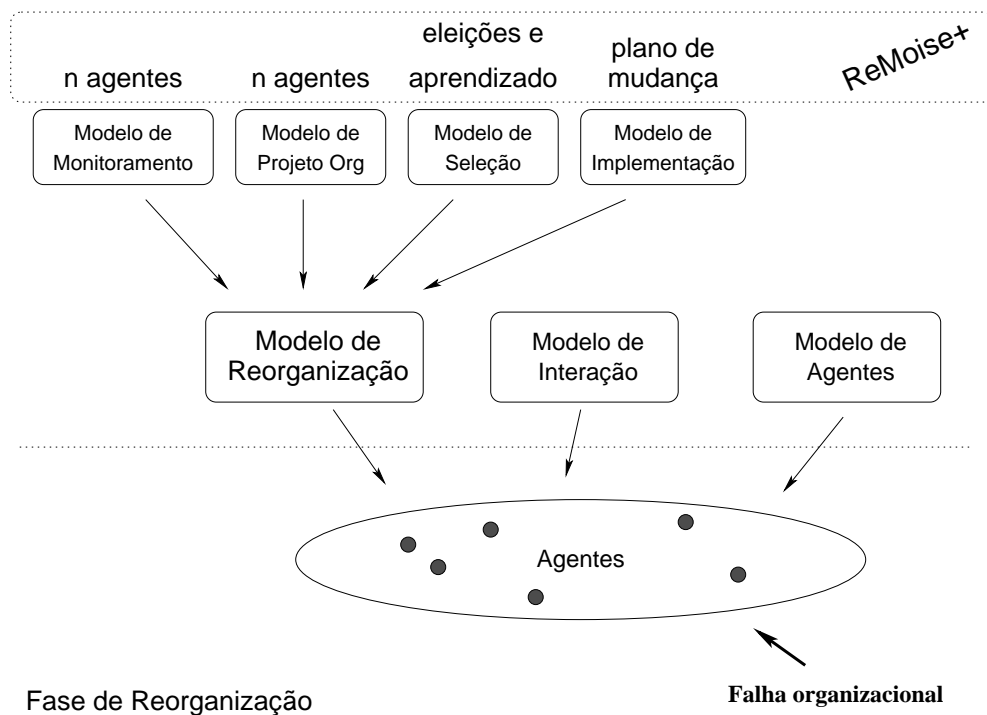


Figura 6.7: Resumo do modelo proposto de reorganização.

- Dois modelos de seleção foram definidos: votação e aprendizado.
- Como modelo de implementação, a noção de plano de reorganização foi desenvolvida.

O detalhamento maior nas duas últimas etapas neste capítulo é justificada pela contribuição que representam, já que não foram encontrados outros trabalhos na área que explorassem estas duas etapas do processo de reorganização.

Comparando esta forma de reorganização com as propostas apresentadas no capítulo 5, tem-se as seguintes vantagens:

1. não somente a entidade organizacional pode ser alterada, mas também sua especificação (Ishida, Gasser e Yokoo (1992), Glaser e Morignot (1997), por exemplo, consideram apenas a mudança na entidade);
2. mais de uma dimensão da organização é considerada na reorganização (Holling, Benyo e Lesser (2001), So e Durfee (1993), por exemplo, consideram somente a dimensão funcional);

3. não há um conjunto fixo de novas organizações (como no caso de Strugeon (1995));
4. a forma de monitoramento/seleção não é estática (como em Horling, Benyo e Lesser (2001) que utiliza um sistema de diagnóstico), mas é aberta;
5. a organização do processo de reorganização é definido da mesma forma que o restante da organização do SMA (nenhuma outra proposta tem esta propriedade); e
6. a etapa de implementação considera um tempo de transição entre a organização corrente e a nova organização por meio do plano de reorganização (novamente, as outras propostas não detalham como acontese a implantação de uma nova organização).

Muitas destas vantagens decorrem da utilização do modelo \mathcal{MOISE}^+ que tem duas funções importantes nesta proposta de reorganização: (*i*) como ferramenta de especificação, permitiu descrever um grupo responsável pela reorganização e seu funcionamento global (os papéis de *Historian* e *Monitored* são um bom exemplo). Assim como o grupo **ReorgGr** descrito aqui, outros poderiam ser descritos com o \mathcal{MOISE}^+ ; e (*ii*) como modelo organizacional utilizado pelo SMA, permite obter várias informações úteis para a reorganização, por exemplo: a finalidade da sociedade, a taxa de sucesso dos planos, a formação dos grupos, etc.

A utilização do próprio modelo \mathcal{MOISE}^+ para especificar o processo de reorganização não determina *como* os agentes irão realizar suas tarefas (como irão projetar novas organizações, por exemplo), mas apenas quais são as restrições sobre o comportamento dos agentes atuantes no processo de reorganização (por exemplo, com quem podem conversar, autoridades, obrigações, etc.). Este fato já era esperado, pois o \mathcal{MOISE}^+ permite a especificação de um aspecto global (a organização) de um sistema e não os aspectos individuais (o funcionamento dos agentes). Assim, se definiu a *organização do processo de reorganização* e não os agentes responsáveis pela reorganização (chamados aqui de agentes da configuração da reorganização). A vantagem desta abordagem é a mesma de um SMA ter uma organização explicitamente definida (a visão OC). Mais precisamente, o processo de reorganização está explicitamente definido e disponível para os agentes em uma notação conhecida por eles. Os agentes podem raciocinar

nar e participar do processo de reorganização, da mesma forma que participam em qualquer outro esquema. O modelo não exclui, portanto, a possibilidade da própria “organização da reorganização” ser alvo de um processo de reorganização, introduzindo um novo nível de reflexividade.

Apesar da proposta apresentada ter propriedades importantes (ser aberta, distribuída, etc.), ainda falta definir, em mais detalhes, como os agentes irão monitorar, projetar e selecionar organizações. Nos estudos de caso, tais detalhes serão melhor esclarecidos.

Parte III

Estudios de caso

7 E-Alliance

Este capítulo tem por objetivo avaliar o processo de reorganização proposto no capítulo 6 em um outro domínio de aplicação: empresas virtuais. Mais precisamente, avaliar o uso de reorganização e do modelo $\mathcal{M}OISE^+$ para solucionar alguns dos problemas abordados pelo projeto E-Alliance (ANDREOLI; CASTELLANI; MUNIER, 2000). Além de verificar a adequação do grupo **ReorgGr** neste novo contexto, pretende-se também detalhar como funcionam alguns dos agentes que realizam a reorganização, principalmente o agente que assume o papel de *Selector*.

7.1 Introdução

O projeto E-Alliance tem por objetivo fornecer o suporte computacional necessário às atividades de colaboração e negociação entre diferentes participantes de uma aliança inter-empresarial (CARRON; BOISSIER, 2001; ANDREOLI; CASTELLANI; MUNIER, 2000). Por exemplo, se um dos participantes recebe uma requisição de um cliente e ele não pode ou não deseja atendê-la sozinho, tal participante pode solicitar (negociar, na verdade) com os demais membros da aliança a realização de parte do trabalho.

Apesar de já existirem trabalhos na área de Business to Business (B2B) que consideram o mesmo tipo de aplicação, o interesse do projeto está em alianças onde as seguintes características são importantes:

- ser aberto: novos participantes podem entrar na aliança;
- não ser intrusivo: a autonomia dos participantes deve ser mantida;
- centrado em negociação: a troca de serviços, produtos ou informações entre

os participantes é estabelecida por meio de negociações, que posteriormente são registradas na forma de contratos;

- tais contratos podem descrever vários tipos de acordos, por exemplo: flexibilidade de prazos; custos por quebra do contrato; grau de controle que o contratante tem sobre a execução de uma tarefa, etc.

O cenário escolhido para exemplificar e validar as propostas do projeto E-Alliance são as alianças entre gráficas de impressão (*Printshops*). Cada gráfica é autônoma e responsável por sua própria carga de trabalho, recursos, clientes, contratos, etc. Apesar desta autonomia, quando uma gráfica aceita participar da Aliança, ela adota certas regras de cooperação com as demais participantes. Particularmente, estas regras servem para facilitar o funcionamento da aliança e manter sua coesão. Por exemplo, uma participante não pode somente enviar tarefas às outras participantes e nunca aceitar tarefas de outras.

É justamente para estabelecer as regras de cooperação entre os participantes que o $MOISE^+$ pode ser útil, isto é, pode-se estabelecer quais são os papéis dos participantes, suas obrigações, suas metas, etc. Uma vez estabelecida uma organização para a Aliança, deve-se considerar também a reorganização desta Aliança pois este domínio de aplicação é bastante dinâmico. Evidentemente, a especificação organizacional descrita aqui para a Aliança atende somente alguns dos requisitos do projeto E-Alliance, isto é, aqueles requisitos que podem ser especificados por uma abordagem organizacional e/ou reorganizacional.

Para iniciar a história da Aliança, é suposta uma primeira EO muito simples que está descrita na figura 7.1. Nesta EO, os agentes podem assumir somente um papel (*Printshop*) que lhes dá direito de comunicação com os demais *Printshops*. Não há nem EF e nem ED para o grupo **alliance**. Esta EO também inclui um sub-grupo **ReorgGr**, responsável pela adaptação organizacional da Aliança. Todos os agentes *Printshop* do grupo **alliance** podem assumir também o papel de *Monitor* do grupo **ReorgGr**. Cinco agentes, com nomes “A”, “B”, “C”, “Sel” e “Org”, adotaram esta EO para sua Aliança criando a EnO da figura 7.2.

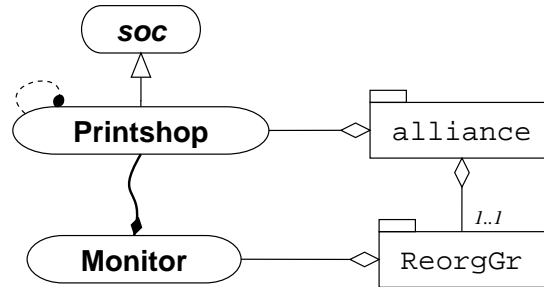


Figura 7.1: Primeira especificação organizacional da Aliança.

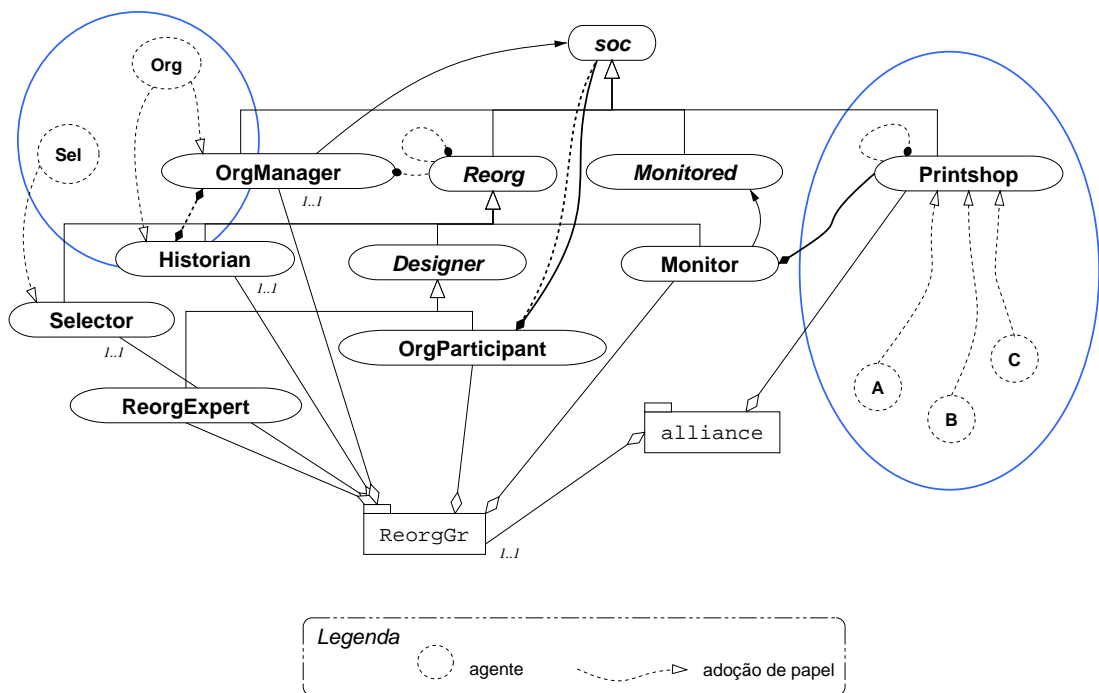


Figura 7.2: Primeira entidade organizacional da Aliança.

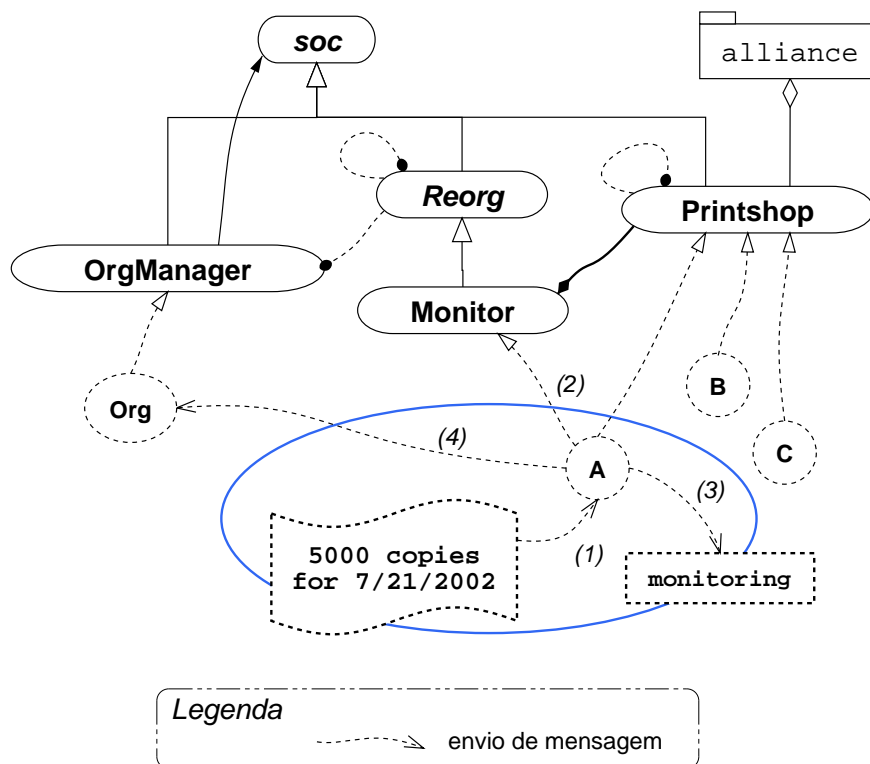
7.2 Primeira reorganização do grupo E-Alliance

Esta seção relata como poderia acontecer a reorganização da Aliança com uma configuração de reorganização que será formada pelos agentes *Org* e *Sel* que desempenham, respectivamente, os papéis de *OrgManager* e *Selector*. Estes dois agentes são de propósito geral (cujo funcionamento já foi apresentado no capítulo 6), sendo que o agente *Sel* utiliza a estratégia de votação para selecionar uma proposta de reorganização. Completando a configuração de reorganização, os papéis de *Monitor* e *Designer* serão desempenhados pelo agente *A* que conhece o domínio de aplicação. O funcionamento deste agente será exemplificado nas sub-seções seguintes.

7.2.1 Monitoração

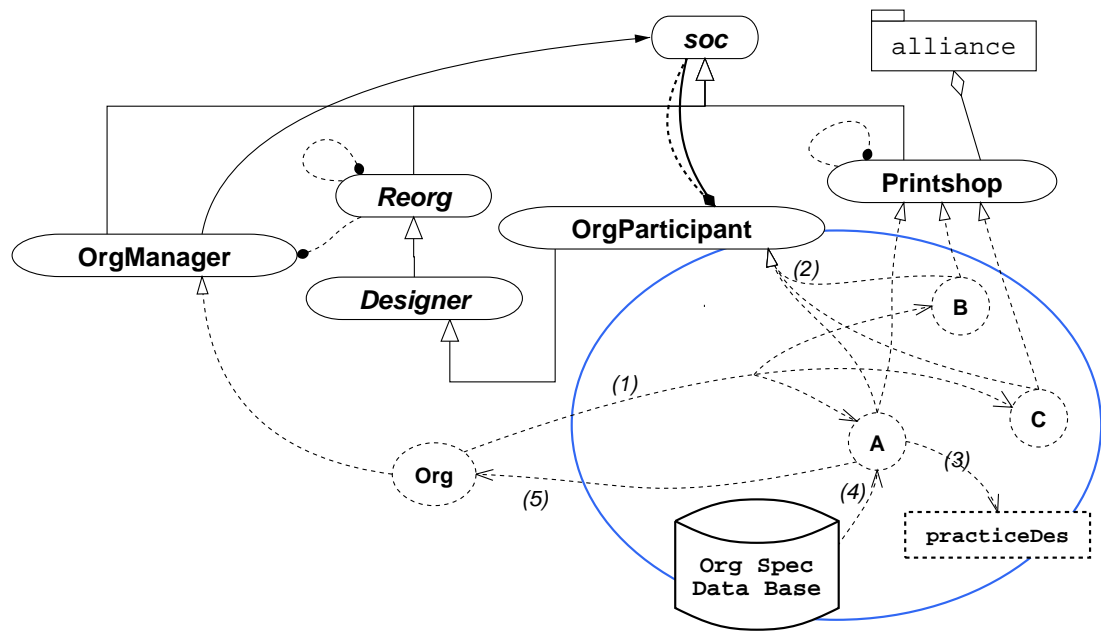
Supondo que o agente *A* recebeu uma requisição para imprimir 5000 cópias de um livro até o dia 21 de julho (figura 7.3). Consultando sua programação de atividades, o agente *A* descobre que ele não pode atender esse pedido sozinho e decide utilizar os serviços de parceiros da Aliança para não perder o pedido. Certamente a organização corrente permite que *A* negocie o serviço com os demais, já que todos os *Printshops* podem se comunicar. Contudo, pode-se perceber que esta EO inicial é muito ampla e não colabora para que a meta de *A* seja satisfeita. Por exemplo, os contratos não estão representados, o agente *A* não tem controle sobre o estado corrente da execução das tarefas que delegou a outros, etc.

Por estas razões, o agente *A* decide alterar a EO corrente. Isto é possível somente por meio do grupo *ReorgGr* e do esquema sch_{reorg} . Conhecendo a EO corrente, particularmente o grupo de reorganização e a compatibilidade com *Monitor*, o agente *A* assume o papel de *Monitor* e, conseqüentemente, a missão m_2 (monitorar). Tal missão lhe permite satisfazer a meta *monitoring*, o que inicia a etapa de projeto de uma nova organização. Como no exemplo 6.1, o *Monitor A* envia uma mensagem para o *OrgManager* com a descrição da falha $Fault = (purposeChanged, EO)$.



Em (1), o agente *A* recebe um pedido para imprimir 5000 livros. Para atender esse pedido, *A* decide mudar a EO e se assume o papel de Monitor (2) e a missão de monitoração (3). Por fim (4), envia uma mensagem para o *Org* indicando a razão da reorganização.

Figura 7.3: Monitoração da Aliança.



Em (1), o *OrgManager* convida os *Printshops* a assumirem o papel de *OrgParticipant*. Os três aceitam o convite (2), mas somente o agente *A* se compromete com a missão *practiceDes* (3). O agente *A* consulta um repositório de EOs (4), escolhe uma alternativa e envia a proposta para o *OrgManager* (5).

Figura 7.4: Projeto de uma nova EO para a Aliança.

7.2.2 Projeto

Iniciada a etapa de projeto, o agente *Org* convida todos os agentes participantes da Aliança para assumirem papéis e missões no processo de reorganização. Todos os três agentes aceitam o papel de *OrgParticipant*. Contudo, somente o agente *A* aceita a missão m_5 (projetar uma nova organização e votar). A figura 7.4 ilustra os passos realizados nesta etapa de projeto.

A fim de satisfazer a meta *practiceDes* (projeto de uma organização a partir do conhecimento adquirido por meio da participação na organização), o agente *A* consulta um repositório de especificações organizacionais e escolhe uma especificação que ele julga adequada para sua meta inicial (imprimir 5000 livros). O agente utiliza o novo objetivo, imprimir livros, como guia na busca por uma nova EO.

Supondo que o *OrgParticipant A* escolheu a EO ilustrada na figura 7.5, que inclui um grupo e um esquema para impressão de livros, ele tem que construir

um plano para mudar a EO corrente para a nova EO. Neste caso, não é necessário remover componentes da EO corrente, apenas adicionar um novo grupo, um esquema e a ligação deontica entre os dois. Portanto, A encaminha como proposta o seguinte plano de mudança:

- foco na EE

% papel *Printshop* passa a ser monitorado.

Comandos de reorganização:

- remover o papel *Printshop*,
- adicionar o papel *Printshop* como sub-papel de *Monitored*.

% criação de um novo grupo: GrForBook

Comandos de reorganização:

- adicionar papel *Manager* como sub-papel de *Printshop*,
- adicionar papel *Printer* como sub-papel de *Printshop*,
- adicionar papel *Binder* como sub-papel de *Printshop*,
- adicionar o grupo GrForBook como sub-grupo de *alliance* e cardinalidade mínimo=0 e máximo=10,
- incluir o papel *Manager* no grupo GrForBook com cardinalidade mínimo=1 e máximo=1,
- incluir o papel *Printer* no grupo GrForBook com cardinalidade mínimo=1 e máximo=5,
- incluir o papel *Binder* no grupo GrForBook com cardinalidade mínimo=1 e máximo=2,
- adicionar, no escopo do GrForBook, uma ligação de autoridade partindo de *Manager* e indo para *Printer*,
- adicionar, no escopo do GrForBook, uma ligação de autoridade partindo de *Manager* e indo para *Binder*,

- adicionar, no escopo do GrForBook, uma relação de compatibilidade simétrica entre os papéis de *Printer* e *Binder*.

- foco na EF

% Criação de um esquema para imprimir livros

Comandos de reorganização:

- adicionar um esquema chamado sch_{pb} tendo como meta raiz *printBooks*,
- adicionar ao esquema o plano
“*printBooks = print(Pqty, Pcost), bind(Bqty, Bcost)*”,
- adicionar ao esquema a missão *mm* com a meta *printBooks* e cardinalidade mínimo=1 máximo=1,
- adicionar ao esquema a missão *mp* com a meta *print* e cardinalidade mínimo=1 máximo=5,
- adicionar ao esquema a missão *mb* com a meta *bind* e cardinalidade mínimo=1 máximo=2.

- foco na ED

% relacionar o grupo GrForBook com o esquema printBooks

Comandos de reorganização:

- adicionar ao papel *Manager* a permissão para a missão *mm*,
- adicionar ao papel *Printer* a obrigação para a missão *mp*, no período *Ptc*, como penalidade para o não cumprimento desta obrigação paga-se $Pcost/3$,
- adicionar ao papel *Binder* a obrigação para a missão *mb*, no período *Btc*, como penalidade para o não cumprimento desta obrigação paga-se $Bcost/4$.

que, se implementado, altera a EO da Aliança como ilustra a figura 7.5. A proposta do agente *A* tem foco nos três aspectos organizacionais (EE, EF e ED). Nas alterações estruturais, propõe-se a criação de uma nova especificação de grupo (**GrForBook**) que será criada pelos agentes no momento de atender um pedido de impressão de livros. Este grupo irá imprimir os livros de acordo com um esquema muito simples mas que permite determinar as atribuições dos agentes que assumirem os papéis de Manager (gerência da impressão dos livros), Printer (impressão de uma quantidade *Pqty* de livros pelo preço *Pcost* na data *Ptc*¹) e Binder (montagem de *Bqty* livros pelo preço *Bcost* na data *Btc*). Esta proposta de EO inclui às relações deonticas atributos particulares do domínio de aplicação. No caso, é incluído um valor a ser pago como penalidade pelos participantes da Aliança caso não cumpram seus papéis.

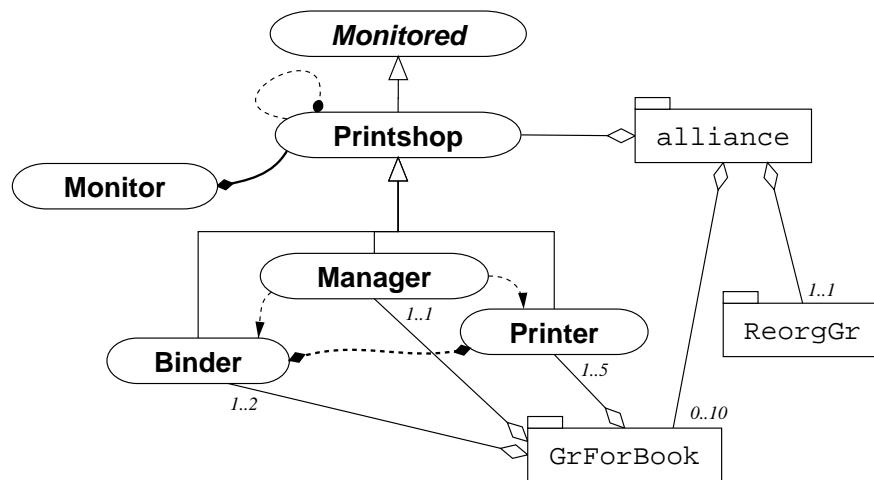
7.2.3 Seleção e Implementação

Como há somente duas propostas (a proposta de *A* e a “proposta” de não mudar nada), o processo de votação, executado pelo *Sel*, é simples. No caso, vamos supor que todos os *Printshops* aceitam a proposta de *A* (na seção 7.3 este resultado é alcançado de forma menos consensual), que é implementada em seguida. Ao término da implementação da nova EO, a EnO da Aliança tem um estado semelhante ao mostrado na figura 7.6. Nota-se que os papéis dos agentes continuam os mesmos de antes da reorganização, pois a implementação de uma nova especificação de grupo não implica que os agentes irão assumir os papéis desta nova especificação.

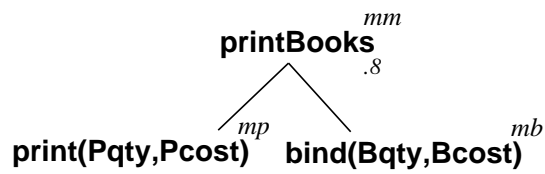
O processo de reorganização termina com a implementação da nova EO. Contudo, para exemplificar como a nova organização pode ser útil para a meta do agente *A* — que motivou a reorganização — será descrito a seguir como os agentes podem utilizar esta nova EO.

Exemplo 7.1 O agente *A* negocia com os outros participantes as condições (preço e prazo) para aceitarem os papéis de *Printer* e *Binder*. Como decorrência desta negociação (não descrita aqui por não ser o objeto da reorganização) o agente *B* aceita realizar os serviços de impressão e montagem de livros para o

¹Os argumentos das metas e das relações (*Pqty*, *Pcost* e *Ptc*) têm seus valores atribuídos na criação de uma instância deste esquema.



(a) EE



(b) EF

papel	relação deôntica	missão	restrições de tempo	penalidade
Manager	<i>per</i>	<i>mm</i>	<i>Any</i>	—
Printer	<i>obl</i>	<i>mp</i>	<i>Ptc</i>	<i>Pcost/3</i>
Binder	<i>obl</i>	<i>mb</i>	<i>Btc</i>	<i>Bcost/4</i>

(c) ED

Figura 7.5: Proposta de alteração enviada pelo agente A.

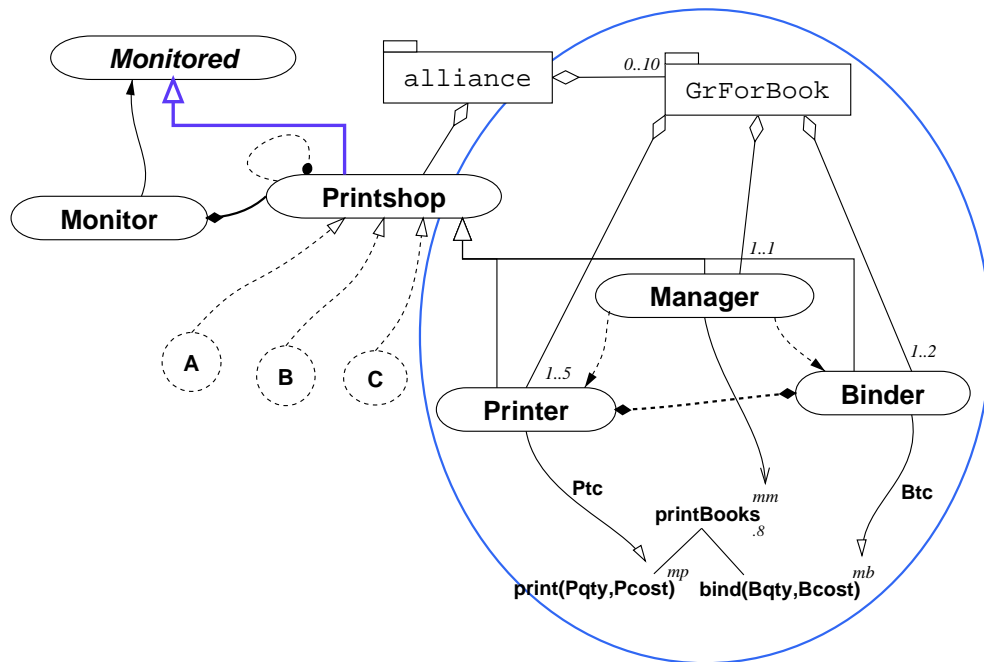


Figura 7.6: Segunda EnO da Aliança.

agente *A*. Para implementar o resultado da negociação no sistema, o agente *A* cria uma instância do grupo *GrForBooks* e assume o papel de *Manager*. O agente *B* assume os papéis *Printer* e *Binder* no novo grupo. Como estes dois papéis são compatíveis, o mesmo agente pode assumi-los. O *Manager* do grupo (*A*) cria em seguida uma instância do esquema para impressão de livros (esquema *pb*). Este agente pode criar o esquema, já que tem permissão (via o papel de *Manager*) para a raiz do esquema. O agente *B* é obrigado a se comprometer com as missões *mp* (imprimir) e *mb* (montar) do novo esquema, pois seus papéis no grupo *GrForBooks* possuem estas obrigações. *A* também deve atribuir os valores das variáveis do esquema conforme a negociação que teve com o agente *B*, por exemplo:

- $Pqty = 5000$: quantidade de livros que o agente *B* se compromete a imprimir,
- $Pcost = 400$: valor pago ao agente *B* pela impressão, implicando em uma penalidade de $400/3$ que *B* deve pagar a *A* se não cumprir a missão de imprimir,
- $Bqty = 5000$: quantidade de livros que o agente *B* se compromete a montar,
- $Bcost = 200$: valor pago ao agente *B* pela montagem, implicando em uma pena-

lidade de 200/4,

- Ptc = “start at 5/20/2002, end at 6/18/2002”: restrições temporais para a missão de impressão e
- Btc = “start at 6/20/2002, end at 7/10/2002”: restrições temporais para a missão de montagem.

Enfim, a utilização da EO deixa bem claro quais as responsabilidades de cada um dos agentes participantes.

A criação do novo grupo e a adoção de novos papéis não extingue o grupo *alliance*, os agentes continuam com o papel de *Printshop*. Contudo, no contexto da execução do esquema pb , os agentes devem se submeter às restrições dos papéis do grupo *GrForBooks*, onde, por exemplo, existem relações de autoridade. \square

Exemplo 7.2 Um outro exemplo com a mesma EO poderia ser o caso onde o agente A assume o papel de *Manager*, o agente B assume o papel de *Printer* e *Binder*, e o agente C também assume o papel de *Binder*. Nesta outra instância do grupo *GrForBooks*, A poderia tanto criar duas instâncias do esquema pb (veja pb^1 e pb^2 na tabela 7.1) como criar somente um esquema (pb^3 da tabela 7.1).

O compromisso com metas como $bind(0,0)$ é necessário pois o papel de *Binder* é obrigado a realizar a missão mb e sua meta $bind$.

\square

No contexto do projeto E-Alliance, a nova EO representa um *modelo de contrato* que facilita o processo de negociação entre os agentes, já que eles sabem que adoção de papel está sendo negociada. Quando instanciado, o contrato determina (i) o comportamento esperado de cada participante — por meio dos papéis e missões — e (ii) o modo como o contrato será executado — por meio da especificação de esquemas.

7.3 Segunda reorganização do grupo E-Alliance

A reorganização descrita nesta seção considera uma configuração de reorganização mais ampla. Além dos agentes *Org*, *Sel* e os membros da Aliança (A , B e C),

Tabela 7.1: Exemplo de adoção de papéis, missões e metas na Aliança

agente	papel	missão	meta
esquema pb^1			
A	Manager	mm	
B	Printer	$mp(Ptc=$ “start at 6/1/2002, end at 6/30/2002”)	$print(2000,300)$
B	Binder	$mb(Btc=$ “start at 6/30/2002, end at 7/10/2002”)	$bind(2000,100)$
C	Binder	mb	$bind(0,0)$
esquema pb^2			
A	Manager	mm	
B	Printer	$mp(Ptc=$ “start at 5/20/2002, end at 6/18/2002”)	$print(3000,400)$
B	Binder	mb	$bind(0,0)$
C	Binder	$mb(Btc=$ “start at 6/20/2002, end at 7/10/2002”)	$bind(3000,150)$
esquema pb^3			
A	Manager	mm	
B	Printer	$mp(Ptc=$ “start at 5/20/2002, end at 6/18/2002”)	$print(5000,700)$
B	Binder	$mb(Btc=$ “start at 6/30/2002, end at 7/10/2002”)	$bind(2000,100)$
C	Binder	$mb(Btc=$ “start at 6/20/2002, end at 7/10/2002”)	$bind(3000,150)$

outros quatro agentes serão incluídos na configuração: um agente de monitoração da organização e três agentes especialistas na elaboração de propostas de reorganização.

7.3.1 Monitoração

Depois da Aliança funcionar por um certo período com a EO resultante da reorganização proposta pelo agente *A*, a Aliança decidiu colocar um novo agente no grupo **Reorg**: um agente especializado em monitorar um tipo particular de esquema. Este novo *Monitor*, chamado *MPB* (Monitor do esquema de impressão de livros *pb*), se compromete com a missão $sch_{reorg}.m_2$ e, portanto, com a meta *monitoring*. Esta meta é satisfeita pelo *MPB* quando a taxa de sucesso do esquema *pb* ficar abaixo de 60%. Para saber a taxa de sucesso do esquema *pb*, basta consultar a EO mantida pelo *OrgManager* que inclui os esquemas e suas taxas de sucesso. No caso do *MPB* identificar que o valor baixou de 60%, ele informa ao *OrgManager* o início de um processo de reorganização com $Fault = (lowSuccess, sch_{pb})$.

7.3.2 Projeto

No caso do esquema de impressão tornar-se ineficiente e o agente *MPB* disparar a reorganização, o *OrgManager*, ao contrário da primeira reorganização, convida três agentes (E_1 , E_2 e E_3) para assumirem o papel de *ReorgExpert* e também convida as gráficas *A*, *B* e *C* para serem *OrgParticipant*.

Considerando, por exemplo, que os agentes E_1 , E_2 e E_3 se comprometeram com a missão m_4 (projeto de organização por especialista), e que os agentes *A* e *B* se comprometeram com a missão m_5 (projeto por prática), as propostas enviadas por estes agentes poderiam ser as seguintes:

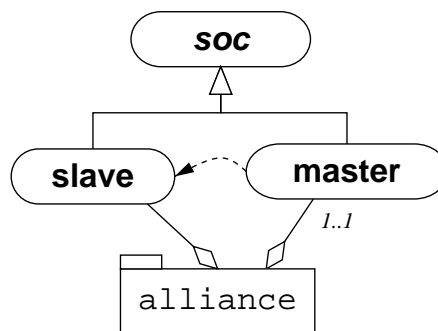
1. O agente E_1 propõe o incremento de 30% nas penalidades da ED corrente (figura 7.7). O argumento deste agente é que o esquema não funciona porque os agentes *Printer* e *Binder* abandonam suas missões muito facilmente.
2. O agente E_2 propõe uma nova e hierárquica EO (figura 7.8). O argumento deste agente é que os agentes *Printer* e *Binder* podem conversar e que o *Manager* não tem muito controle na realização das tarefas.
3. O agente E_3 propõe a divisão do papel de *Printer* em dois (figura 7.9). Seu argumento é que a impressão deve ser separada em impressão colorida e preto e branco.
4. O agente *A*, dada sua experiência no esquema sendo revisado, propõe o aumento do número de Binders (de 2 para 5). Seu argumento é que o grupo atual permite mais Printers que Binders e isso atrasa a impressão.
5. O agente *B*, dada sua experiência como *Printer*, propõe aumentar os detalhes do esquema de impressão (figura 7.10). O argumento é que o *Manager* precisa saber melhor como os *Printers* e *Binders* estão realizando suas tarefas.

7.3.3 Seleção

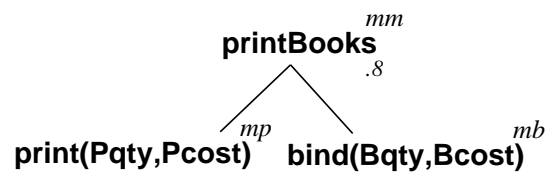
Para classificar as propostas, o agente *Sel* utiliza os seguintes parâmetros: $\theta = 0,5$ (a participação em versões anteriores tem a metade de valor de experiências

papel	relação deôntica	missão	restrições temporais	penalidade
<i>Manager</i>	<i>per</i>	<i>mm</i>	<i>Any</i>	—
<i>Printer</i>	<i>obl</i>	<i>mp</i>	<i>Ptc</i>	$(Pcost/3) * 1.3$
<i>Binder</i>	<i>obl</i>	<i>mb</i>	<i>Btc</i>	$(Bcost/4) * 1.3$

(a) ED

Figura 7.7: Proposta de reorganização do agente E_1 com foco na ED.

(a) EE

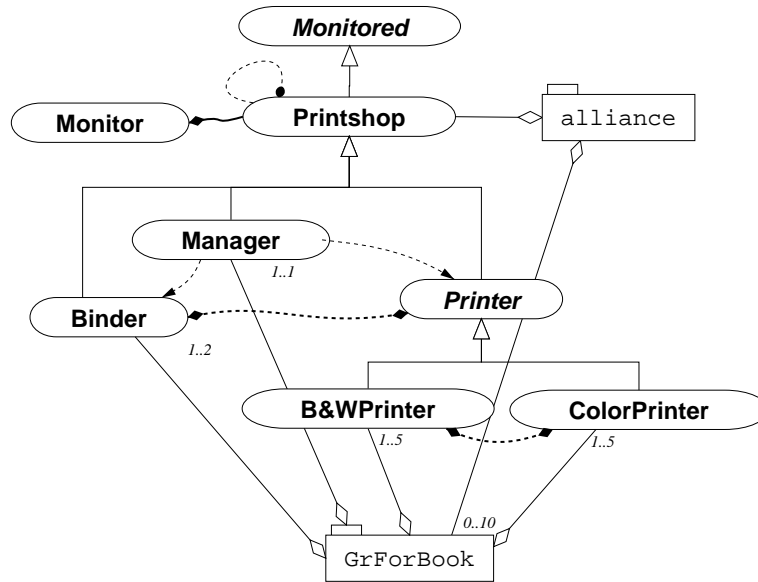


(b) EF

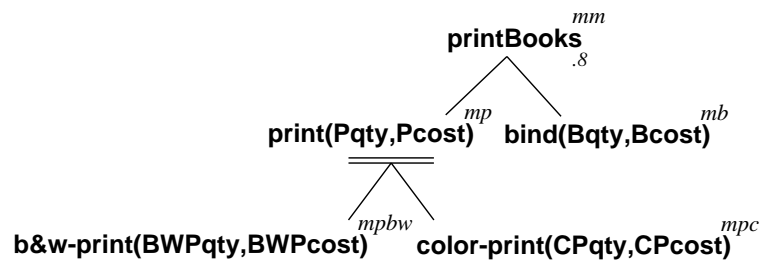
papel	relação deôntica	missão	restrições temporais	penalidade
<i>Master</i>	<i>per</i>	<i>mm</i>	<i>Any</i>	—
<i>Slave</i>	<i>obl</i>	<i>mp, mb</i>	<i>TC</i>	$Pcost/2$

(c) ED

Figura 7.8: Proposta de reorganização do agente E_2 com foco na EO.



(a) EE



(b) EF

papel	relação deontica	missão	restrições temporais	penalidade
Manager	<i>per</i>	<i>mm, mp</i>	<i>Any</i>	—
B&WPrinter	<i>obl</i>	<i>mpbw</i>	<i>PbwTC</i>	<i>BWPcost/3</i>
ColorPrinter	<i>obl</i>	<i>mpc</i>	<i>PcTC</i>	<i>CPcost/3</i>
Binder	<i>obl</i>	<i>mb</i>	<i>BTC</i>	<i>Bcost/4</i>

(c) ED

Figura 7.9: Proposta de reorganização do agente E_3 com foco na EO.

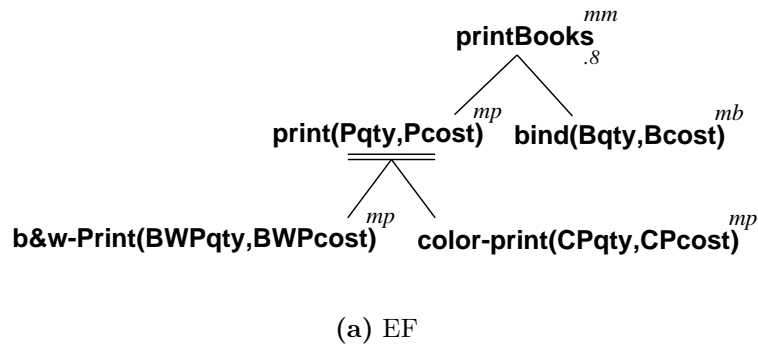


Figura 7.10: Proposta de reorganização do agente *B* com foco na EF.

na versão corrente da EO), $\epsilon = 0,05$ (cada dia que uma EO ficou ativa conta 0,05) e $\phi = 2$ (o peso do custo de mudança é 2). A classificação das propostas considera que o *Historian* registrou a história da tabela 7.2 para a Aliança, da qual as participações resumidas na tabela 7.3 podem ser deduzidas e utilizadas para calcular a experiência dos agentes.

Tabela 7.2: Resumo da história da entidade organizacional Aliança

dia	evento	EO
1/2/2002	os_1 (com a especificação da figura 7.1) iniciou	os_1
	os agentes <i>A</i> , <i>B</i> e <i>C</i> assumiram o papel de <i>Printshop</i>	os_1
1/5/2002	a proposta de mudança enviada pelo agente <i>A</i> (figura 7.5) é implementada	os_2
	<i>A</i> teve papel <i>Manager</i> e <i>B</i> papéis <i>Printer</i> e <i>Binder</i> em duas execuções do esquema de impressão de livros <i>pb</i>	os_2
	<i>B</i> teve papel <i>Manager</i> , <i>A</i> papel <i>Printer</i> e <i>C</i> papel <i>Binder</i> em uma execução do esquema <i>pb</i>	os_2
	<i>A</i> teve papel <i>Manager</i> , <i>B</i> papel <i>Printer</i> e <i>Binder</i> , e <i>C</i> papel <i>Binder</i> em duas execuções do esquema <i>pb</i>	os_2
1/10/2002	<i>C</i> tem papel <i>Manager</i> , <i>A</i> papel <i>Printer</i> e <i>B</i> papel <i>Binder</i> em uma execução do esquema <i>pb</i> . Portanto, cada um dos agentes tem, no momento, uma missão e dois papéis (um no grupo <i>GrForBook</i> e outro no grupo <i>alliance</i>)	os_2
1/10/2002	os_2 terminou pois iniciou-se uma nova reorganização iniciou. A versão 2 da EO ficou 153 dias ativa	

Um vez publicada a classificação (tabela 7.4), a eleição poderia apresentar os resultados mostrados na tabela 7.5.² Portanto, a proposta do agente E_3 será

²As razões que levaram os agentes a votar em determinada proposta não foram consideradas

Tabela 7.3: Papéis e missões dos agentes da Aliança

agente	quantidade papéis em alliance	em GrForBook	comprometimento com missões de sch_{pb}
<i>A</i>	em os_1		
	1 × PrintShop		
	em os_2		
	1 × PrintShop	2 × Manager 2 × Printer	4 × mm 2 × mp
<i>B</i>	em os_1		
	1 × PrintShop		
	em os_2		
	1 × PrintShop	1 × Manager 2 × Printer 3 × Binder	1 × mm 4 × mp 5 × mb
<i>C</i>	em os_1		
	1 × PrintShop		
	em os_2		
	1 × PrintShop	1 × Manager 2 × Binder	1 × mm 3 × mb

Tabela 7.4: Classificação das propostas

proposta	foco	fe	da	$cost$	$position$
E_1	ED	$0,5*0+0+0=0$	0	$2*(0+0)=0$	0
E_2	EO	$0,5*0+0+0=0$	0	$2*(6+3)=18$	-18
E_3	EO	$0,5*0+0+0=0$	0	$2*(1+1)=4$	-4
<i>A</i>	EE, $gr_{GrForBook}$	$0,5*1+4+0=4,5$	7,65	$2*(0+0)=0$	12,15
<i>B</i>	EF, sch_{pb}	$0,5*0+0+10=10$	0	$2*(0+1)=2$	8
no change		0	0	0	0

implementada pelo *OrgManager*.

7.4 Conclusões

Como já foi observado na introdução deste capítulo, a utilização do modelo $MOISE^+$ não resolve todos os problemas do projeto E-Alliance, notadamente os problemas do processo de negociação. Entretanto, o $MOISE^+$ mostrou-se útil para:

neste estudo de caso. O modelo somente apresenta uma classificação das propostas, cabendo ao projetista dos agentes determinar como eles votam.

Tabela 7.5: Resultado das eleições

proposta	eleitor	ge do eleitor	da	valor do voto ($ge+da+1$)
E_3	E_1	0	0	1,00
	E_2	0	0	1,00
	E_3	0	0	1,00
	B	$0,5*1+7+10$	0	18,50
	C	$0,5*1+4+4$	0	9,50
	<i>total</i>			<i>31,00</i>
A	A	$0,5*1+5+6$	7,65	20,15
	<i>total</i>			<i>20,15</i>

1. representar a organização da Aliança e desta forma simplificar e direcionar o comportamento dos participantes da Aliança para aqueles voltados às metas comuns;
2. representar parte dos contratos que servem de objeto de negociação inicial dos agentes (a organização que será adotada na execução de um contrato faz parte dele e da sua negociação);
3. representar, na forma de instâncias de grupos e esquemas, os contratos firmados, isto é, as autoridades, objetivos, prazos, etc.
4. viabilizar, por meio da reorganização, a evolução e adaptação da Aliança e seus modelos de contrato a novos ambientes e objetivos.

Ao lado de mostrar a utilidade que o modelo de reorganização tem para o projeto E-Alliance, este estudo de caso ilustra a viabilidade e a abrangência do modelo em um sistema real. A abrangência da modelagem vai até o ponto onde características dependentes do domínio precisam ser incluídas, por exemplo, a elaboração das propostas de reorganização. Nestes pontos, o modelo define papéis e missões que os agentes do domínio devem assumir.

8 JOJTEAM

Este capítulo descreve um segundo estudo de caso que utiliza a proposta de organização/reorganização desta tese. O que o distingue do projeto E-Alliance é o domínio de aplicação: futebol de robôs. Este novo domínio permite demonstrar a abrangência da proposta em áreas com perfis bastante distintos. Uma das características que levou à escolha deste segundo domínio foi a existência de ferramentas de simulação, viabilizando a utilização de aprendizado por reforço no processo de reorganização.

8.1 Introdução

O uso de futebol de robôs como ambiente de testes na área de IA, e particularmente na área de SMA, tem se tornado comum devido aos novos desafios que apresenta (HIROAKI et al., 1997). Ao contrário de ambientes de teste anteriores (como os jogos, o xadrez¹, por exemplo), várias sub-áreas da IA (a robótica e a lógica, por exemplo) podem (e devem) ser utilizadas em conjunto. A área de SMA tem neste domínio a maioria das propriedades de um problema adequado ao uso de SMA (distribuição, controle descentralizado, autonomia e reatividade dos agentes, necessidade de coordenação, ...). Além desta adequação entre técnica e problema, a configuração do jogo (tamanho do campo, número de jogadores, regras de jogo, etc.) é definida por uma instituição, viabilizando assim a realização de competições entre os times e sua consequente avaliação. Além destas propriedades, dado o ambiente dinâmico do futebol de robôs, o time precisa se adaptar às mudanças no ambiente (o time oponente).

Dentre as categorias do futebol de robôs, este estudo de caso é realizado na categoria de robôs pequenos, onde cada time tem cinco robôs que jogam em um

¹Um desafio que, de certa forma, a IA já superou.



Figura 8.1: Tela do simulador TeamBots.

campo de 152,5 cm por 274 cm. Para simplificar o desenvolvimento do estudo de caso e dispensar os recursos de hardware, foi utilizado um simulador desta categoria, o TeamBots (BALCH, 2000) (figura 8.1).

O restante deste capítulo está assim organizado: a próxima seção descreve o funcionamento de um time, chamado de JOJTEAM, organizado segundo o modelo $MOISE^+$. A seção 8.3 descreve como são os agentes que formam a configuração de reorganização do time e a seção 8.3.4 apresenta os resultados do time utilizando o processo de reorganização.

8.2 Desenvolvimento do JOJTEAM

O funcionamento do time será detalhado por meio de sua organização e do funcionamento dos agentes.

8.2.1 Organização do time

A organização do JOJTEAM não é fixa, pois durante um jogo várias organizações podem ser utilizadas. Contudo, a fim de explicar o funcionamento do time, será considerada uma única organização (na seção 8.3, a mudança da organização será incluída no time). A figura 8.2 ilustra esta organização, a especificação mais detalhada é encontrada no anexo B.2.

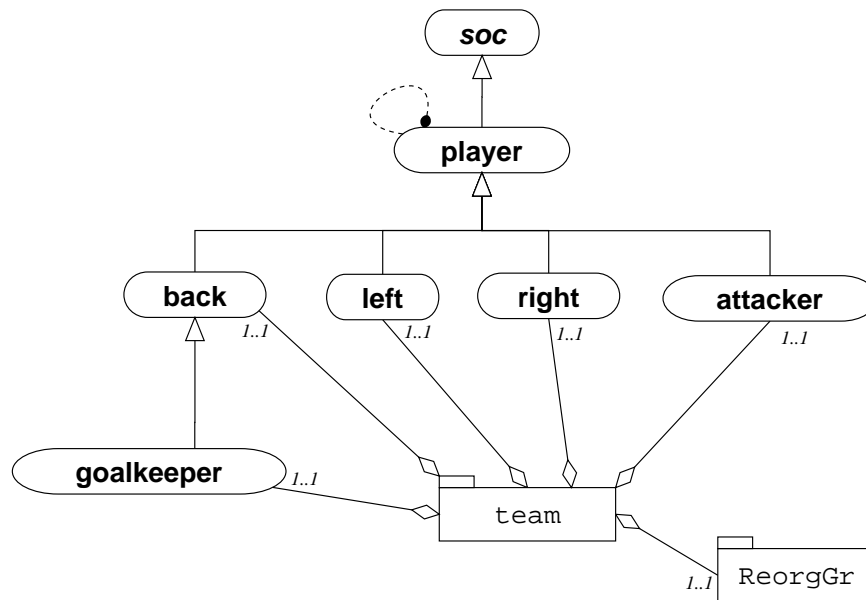
A EE do time é formada por cinco papéis. Cada papel deve ser assumido por um agente, que pode se comunicar com os outros. Existe ainda um sub-grupo de reorganização que será descrito na seção 8.3. Além destas propriedades, nesta aplicação, adicionou-se um atributo particular a cada papel: a área do campo que o agente que assume tal papel deve preferencialmente ficar. Na figura 8.3, há um exemplo de distribuição das cinco áreas.² Por exemplo, o papel de *goalkeeper* tem uma área de atuação próxima à trave de seu time, *left* tem como área a parte esquerda do campo, e *attacker* tem o campo do time oponente como área.

A EF do time tem um único esquema muito simples com o objetivo de fazer gols. Este esquema tem um plano para satisfazer a meta de fazer gols que é composto pela execução paralela das metas: chutar para o gol oponente (*kickToGoal*), defender o próprio gol (*defendOurGoal*) e carregar a bola em direção ao gol oponente e, somente quando próximo ao gol, chutar em direção ao gol oponente (*carryBallToGoal*). Um esquema mais complexo de EF é proposto em (HÜBNER; SICHMAN; BOISSIER, 2002b).

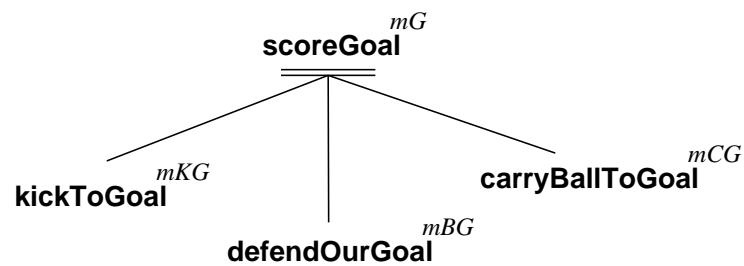
A ED determina quais são os papéis (e conseqüentemente, os agentes) que devem se comprometer com a execução das metas da EF. No caso, o *back* deve procurar chutar a bola em direção ao gol do oponente; *left*, *right* e *attacker* devem carregar a bola em direção ao gol oponente; e *goalkeeper* deve defender o gol do seu time.

Resumidamente, nesta organização do JOJTEAM, é associada a cada papel uma área de atuação e uma meta apropriada para esta área. No funcionamento do time, são criados cinco agentes e cada um deles assume um dos papéis acima (de acordo com sua posição inicial no campo) e se compromete com a meta

²Apesar do modelo *MOISE+* não possuir este atributo nos papéis, sua implementação permite que sejam adicionados aos papéis atributos específicos do domínio de aplicação. No exemplo do anexo B.2, observa-se como esta inclusão é realizada.



(a) estrutura



(b) funcionamento

papel	relação deôntica	missão	restrições de tempo
<i>back</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>left</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>right</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>attacker</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>goalkeeper</i>	<i>obl</i>	<i>mBG</i>	<i>Any</i>

(c) relações deônticas

Figura 8.2: Especificação organizacional do JOJTEAM.

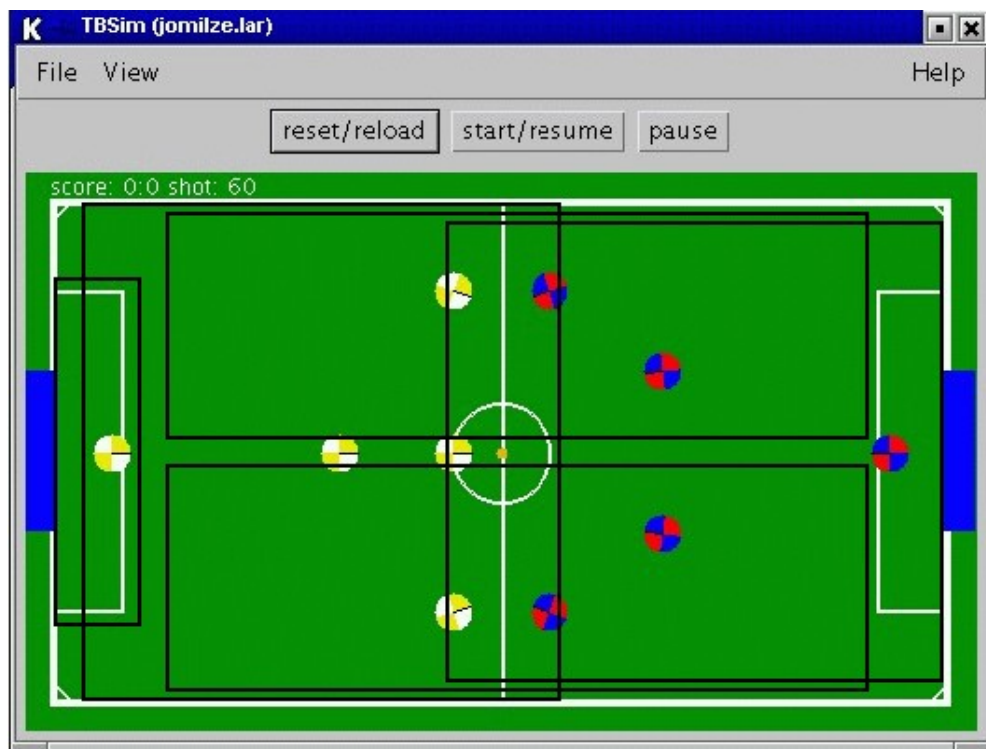


Figura 8.3: Exemplo de área dos jogadores.

adequada. Tendo uma missão, cada agente sabe interpretar e executar a meta da missão. A seção seguinte explica como este processo ocorre internamente no agente.

8.2.2 Funcionamento dos agentes

A arquitetura geral do agente, ilustrada na figura 8.4, é formada pelas seguintes camadas:

Camada organizacional: cada agente jogador possui um OrgBox que lhe permite conhecer a organização e suas metas (conforme descrito no capítulo 4).

Camada deliberativa: esta camada, no caso dos jogadores, é muito simples pois considera um agente com um único papel, uma única missão e uma única meta. Conforme especificado no algoritmo da figura 8.5, o agente decide adotar um papel caso ainda não tenha algum, depois se compromete com a missão que tal papel lhe obriga. Tendo uma missão, por meio do OrgBox, obtém uma meta global para satisfazer (por exemplo, as metas do

esquema da figura 8.2). Como se pode perceber, o agente não tem metas pessoais. Sua única função é satisfazer uma meta global.

Camada reativa: define o comportamento do jogador no ambiente (para onde virar, correr, chutar, etc.). Este comportamento é determinado a partir da meta global escolhida pela camada deliberativa. Esta camada é implementada associando-se a meta corrente do agente a um esquema motor desenvolvido segundo técnicas de campos de potência (ARKIN, 1998; MURPHY, 2000). Por exemplo, o esquema motor da meta *defendOurGoal* é formado pela composição de outros esquemas motores:

- a) ir em direção a sua área do campo se estiver fora dela (a área de agente é um atrator na terminologia de campos de potência);
- b) não bater nos outros robôs (vários repulsores); e
- c) estar entre a bola e a sua trave (o ponto que é o meio da reta entre a bola e a trave é um atrator).

Como cada sub-esquema motor atua como uma força (representada por um vetor) direcionando o agente em um sentido diferente, a direção final é dada pela composição ponderada dos três sub-esquemas (a soma ponderada dos três vetores). Por exemplo, esquema (a) tem peso 7, o esquema (b) tem peso 1 e o esquema (c) tem peso 1,2.

Esta arquitetura, mesmo que simples, atende às necessidades de um agente que suporte reorganizações. No caso, mudanças organizacionais refletem diretamente na camada reativa do agente. Por exemplo, caso a EF mude (trocar o esquema de fazer gols, por exemplo), a missão do agente no esquema antigo é perdida pois o esquema não existe mais (a camada organizacional mantém o conjunto de missões do agente atualizado segundo a EO e a EnO). Estando sem missão, a camada deliberativa escolhe uma nova missão conforme definido no novo esquema. A nova missão, por sua vez, define uma nova meta para a camada reativa.

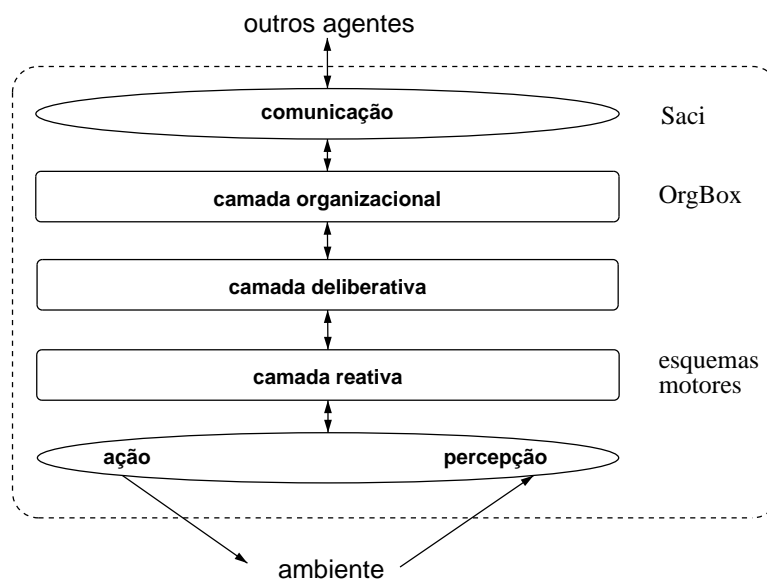


Figura 8.4: Arquitetura geral para o agente jogador do JOJTEAM.

8.3 Configuração de reorganização com aprendizado

A reorganização do JOJTEAM é realizada pelo *ReorgGr*. Os agentes que participam deste grupo não são os jogadores, mas agentes especializados nas tarefas de monitorar, projetar e selecionar organizações. Estes agentes formam a configuração de reorganização.

8.3.1 Monitoração

A monitoração do JOJTEAM é feita de uma forma singular. A cada 24.000 passos de simulação, o agente de monitoração inicia uma reorganização. Como o jogo tem 120.000 passos, existem 5 processos de reorganizações em cada jogo, independente do placar do jogo. A primeira reorganização acontece no passo 0, a segunda no passo 24.000, e a última no passo 96.000. Deste modo, a reorganização sempre acontece, mesmo que o time esteja jogando bem. Como consequência, a reorganização pode explorar novas, e eventualmente melhores, organizações. Esta forma de monitoração é atrelada à estratégia do agente de seleção, que será o aprendizado por reforço. Na descrição do agente de seleção, seção 8.3.3, esta forma de monitoração será melhor justificada.

procedure deliberativeLayer

ρ : o papel corrente do agente (obtido da camada organizacional)

m : a missão corrente do agente (obtida da camada organizacional)

g : a meta corrente do agente

if não tem papel na EnO **then**

 adotar um dos cinco papéis da EE conforme minha posição inicial no campo

end if

if tenho papel e não tenho missão **then**

 se comprometer com a primeira missão de `OrgBox.getObligatedMissions()`

end if

if tenho missão e não tenho meta **then**

$g \leftarrow$ a primeira meta de `OrgBox.getPossibleGoals()`

end if

Figura 8.5: Camada deliberativa do agente jogador.

8.3.2 Projeto

Os planos de reorganizações são desenvolvidos pelos seguintes agentes:

Projetista 1x1x3 apresenta um plano de mudança com

- foco na EE: remove a formação corrente do time e cria uma nova formação com 1 *goalkeeper*, 1 *back* e 3 *attackers*. As áreas destes papéis serão:

papel	posição na formação
<i>goalkeeper</i>	próximo à trave do seu time
<i>back</i>	no campo de defesa
<i>attacker-1</i>	no campo de ataque, no centro
<i>attacker-2</i>	no campo de ataque, a esquerda
<i>attacker-3</i>	no campo de ataque, a direita

- foco na ED: define as seguintes obrigações:

papel	relação deontica	missão	restrições de tempo
<i>back</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>attacker</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>goalkeeper</i>	<i>obl</i>	<i>mBG</i>	<i>Any</i>

nas quais todos irão chutar para o gol oponente, exceto o goleiro.

O algoritmo utilizado por este agente para construir esta proposta está especificado na figura 8.6. Os comandos do plano de reorganização possuem uma sintaxe definida e conhecida pelo OrgManager (que irá implantar o plano). Na figura, entretanto, tal sintaxe não é utilizada para facilitar a leitura.

Projetista 1x3x1 apresenta um plano de mudança com

- foco na EE: remove a formação corrente do time e cria uma nova formação com 1 *goalkeeper*, 1 *back*, 1 *left*, 1 *right* e 1 *attacker*. As áreas destes papéis serão:

papel	posição na formação
<i>goalkeeper</i>	próximo à trave do seu time
<i>back</i>	no campo de defesa
<i>left</i>	no lado esquerdo do campo
<i>right</i>	no lado direito do campo
<i>attacker</i>	no campo de ataque

- foco na ED: define as seguintes obrigações:

papel	relação deontica	missão	restrições de tempo
<i>back</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>left</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>right</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>attacker</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>goalkeeper</i>	<i>obl</i>	<i>mBG</i>	<i>Any</i>

que é a mesma proposta deontica do projetista 1x3x1.

Projetista 4x1 apresenta um plano de mudança com

- foco na EE: remove a formação corrente do time e cria uma nova formação com 1 *goalkeeper*, 1 *back*, 1 *left*, 1 *right* e 1 *attacker*. As áreas destes papéis serão:

papel	posição na formação
<i>goalkeeper</i>	próximo à trave do seu time
<i>back</i>	no campo de defesa
<i>left</i>	no campo de defesa a esquerda
<i>right</i>	no campo de defesa a direita
<i>attacker</i>	no campo de ataque

```

procedure buildReorgPlan

                                                                 % foco na EE
if não existe a definição do papel back then
    incluir no plano “criar o papel back como sub-papel de player”
end if
incluir no plano “definir a propriedade ‘area’ de back como ‘-137x40 10x-40’ ”
                                                                 % o centro do campo é o ponto 0x0 e
                                                                 % área é delimitada pelo ponto superior esquerdo e o ponto inferior direito

if não existe a definição do papel goalkeeper then
    incluir no plano “criar o papel goalkeeper como sub-papel de back”
end if
incluir no plano “definir a propriedade ‘area’ de goalkeeper como ‘-137x25 -125x-25’ ”

if não existe a definição do papel attacker1 then
    incluir no plano “criar o papel attacker1 como sub-papel de player”
end if
incluir no plano “definir a propriedade ‘area’ de attacker1 como ‘20x40 137x-40’ ”

if não existe a definição do papel attacker2 then
    incluir no plano “criar o papel attacker2 como sub-papel de player”
end if
incluir no plano “definir a propriedade ‘area’ de attacker2 como ‘-10x76 137x10’ ”

if não existe a definição do papel attacker3 then
    incluir no plano “criar o papel attacker3 como sub-papel de player”
end if
incluir no plano “definir a propriedade ‘area’ de attacker3 como ‘-10x-10 137x-76’ ”

incluir no plano “remover todos os papéis do grupo team”
incluir no plano “adicionar o papel goalkeeper no grupo team”
incluir no plano “adicionar o papel back no grupo team”
incluir no plano “adicionar o papel attacker1 no grupo team”
incluir no plano “adicionar o papel attacker2 no grupo team”
incluir no plano “adicionar o papel attacker3 no grupo team”

                                                                 % foco na EF
incluir no plano “remover todas as obrigações”
incluir no plano “adicionar a missão mBG como obrigação de goalkeeper”
incluir no plano “adicionar a missão mKB como obrigação de back”
incluir no plano “adicionar a missão mKB como obrigação de attacker1”
incluir no plano “adicionar a missão mKB como obrigação de attacker2”
incluir no plano “adicionar a missão mKB como obrigação de attacker3”

```

Figura 8.6: Exemplo de algoritmo para elaboração de plano de reorganização.

- foco na ED: define as seguintes obrigações:

papel	relação deontica	missão	restrições de tempo
<i>back</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>left</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>right</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>attacker</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>goalkeeper</i>	<i>obl</i>	<i>mBG</i>	<i>Any</i>

que é a mesma proposta deontica do projetista 1.

Projetista Random apresenta um plano de mudança com

- foco na EE: remove a formação corrente do time e cria uma nova formação com 5 players. A área deste papel é todo o campo.
- foco na EF: define um novo esquema para realizar gol onde há somente uma meta *randomMove*, mover-se aleatoriamente e chutar a bola se ela estiver próxima. Criar também uma missão com esta meta, a missão *mRd*.
- foco na ED: define a seguinte obrigação:

papel	relação deontica	missão	restrições de tempo
<i>player</i>	<i>obl</i>	<i>mRd</i>	<i>Any</i>

Certamente esta não é uma boa proposta de reorganização em qualquer caso. Contudo, este projetista foi incluído para se verificar se suas propostas não eram selecionadas pelo processo de reorganização.

Projetista Flex apresenta um plano de mudança com

- foco na EE: aumenta em 20 % a área de cada papel existente.

Projetista UnFlex apresenta um plano de mudança com

- foco na EE: diminui em 20 % a área de cada papel existente.

Projetista FlexGolie apresenta um plano de mudança com

- foco na EE: aumenta em 20 % a área do papel goalkeeper.

Projetista UnFlexGolie apresenta um plano de mudança com

- foco na EE: diminui em 20 % a área do papel goalkeeper.

Projetista newDS apresenta um plano de mudança com

- foco na ED: define as seguintes obrigações:

papel	relação deôntica	missão	restrições de tempo
<i>back</i>	<i>obl</i>	<i>mKG</i>	<i>Any</i>
<i>left</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>right</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>attacker</i>	<i>obl</i>	<i>mCG</i>	<i>Any</i>
<i>goalkeeper</i>	<i>obl</i>	<i>mBG</i>	<i>Any</i>

No plano deste projetista, os agentes *left*, *right* e *attacker* carregam a bola até perto do goal oponente e depois chutam. Os demais agentes têm as mesmas obrigações dos outros projetistas.

8.3.3 Seleção

O agente de seleção utiliza a técnica de aprendizado por reforço, como descrito na seção 6.5.2, para aprender a selecionar as propostas enviadas pelos 9 projetistas descritos na seção anterior. Para compreender o funcionamento deste agente, primeiro são especificados os componentes do MDP:

- O conjunto de estados (O) representa os estados do jogo. Cada estado do jogo é descrito pela tupla:

$$\langle tempo, placar \rangle \quad (8.1)$$

onde *tempo* é uma marcação de tempo no jogo e *placar* a diferença de gols. O tamanho do conjunto de estados influencia o desempenho do algoritmo de aprendizado, portanto os valores para *tempo* e *placar* foram limitados. Em relação ao tempo de jogo, $tempo \in \{0, 1, 2, 3, 4\}$, o jogo tem portanto 5 momentos relevantes para o processo de reorganização, sendo $tempo = 0$ o início do jogo. Estes cinco tempos coincidem com os 5 momentos que a monitoração inicia uma reorganização. Quanto ao placar, $placar \in \{-2, -1, 0, 1, 2\}$, os valores indicam a diferença de gols do placar,

placar negativo significa que o time está perdendo. Caso o time esteja perdendo por mais de dois gols de diferença, a variável *placar* recebe -2 e, caso esteja ganhando por mais de 2 gols, $placar = 2$. Com estas restrições aos valores da tupla, o conjunto de estados tem 25 tuplas ($\#O = 25$).

- O conjunto de ações A é formado pelo nome dos 9 projetistas mais a proposta “no change” de não mudar a organização ($\#A = 10$). As ações possíveis podem então ser parafraseadas como “selecionar a proposta do projetista a ”. Esta opção também é justificada pelo desempenho do algoritmo de aprendizado, que é melhor quanto menor for o conjunto de ações. A alternativa de utilizar as propostas (e não os projetistas) para formar o conjunto A tornaria o conjunto muito maior pois o número de propostas possíveis é maior que o de projetistas³.

A função Q , objeto de aprendizado do Q-Learning, tem portanto um conjunto domínio ($O \times A$) com 250 elementos.

- A função de recompensa $r(o, a)$ é dada pelo número de gols marcados menos os gols recebidos no período entre a seleção da proposta a no estado o e o início de uma nova reorganização. A esta diferença de gols é acrescido um custo de reorganização (com o valor, empiricamente estabelecido, de $-0,2$) no caso de não ter sido escolhida a proposta “no change” na última reorganização.

Com esta modelagem, o agente de seleção irá aprender qual o melhor projetista a ser ouvido em cada estado do jogo (tempo e placar), ou seja, os valores de Q . Ele pode, por exemplo, aprender que no início do jogo é melhor escutar a proposta do projetista 1x1x3, depois escutar o projetista newDS e, se perto do final do jogo estiver ganhando, escutar a proposta do projetista 4x1.

Durante a simulação de vários jogos, o agente de seleção aprende, resumidamente, da seguinte forma:

1. Aguarda uma reorganização (que ocorre em um dos cinco tempos possíveis), *tempo* é o tempo desta reorganização.

³Alguns projetistas apresentam sempre a mesma proposta (como o projetista 1x3x1). Contudo, outros projetistas alteram a organização corrente (como o projetista que aumenta a área dos jogadores) e têm, portanto, um conjunto de propostas maior.

2. Se $tempo \neq 0$ (não é a primeira reorganização do jogo)

- (a) A variável o representa a descrição do estado do mundo na última reorganização (em $tempo - 1$) e a é a decisão tomada naquela reorganização.
- (b) A recompensa $r(o, a)$ será a diferença entre o placar na última reorganização e o placar atual. Convém notar que apesar do estado do mundo ter o valor de $placar$ entre -2 e 2 , na função de recompensa este valor não está limitado a esta faixa de valores.
- (c) Atualiza o valor Q conforme o algoritmo da figura 6.5

$$Q(o, a) \leftarrow Q(o, a) + \alpha \left(r(o, a) + \gamma \max_{a'} Q(o', a') - Q(o, a) \right)$$

3. Verifica o placar do jogo e monta o estado corrente do mundo $\langle tempo, placar \rangle$

4. Escolhe uma ação, isto é, escolhe um dos projetistas e conseqüentemente sua proposta.

Esta escolha da ação, nesta aplicação, é dada pela função

$$\epsilon - greedy : O \rightarrow A,$$

definida por Sutton e Barto (1998) como

$$\epsilon - greedy(o) = \begin{cases} \text{random action from } A & \text{se } rv < \epsilon \\ \arg \max_a Q(o, a) & \text{caso contrário} \end{cases} \quad (8.2)$$

onde a função retorna uma ação aleatória se um valor gerado aleatoriamente ($0 < rv \leq 1$) for menor que o fator de exploração ϵ ($0 \leq \epsilon \leq 1$). Caso contrário, retorna a ação com maior estimativa de recompensa. No primeiro caso, a função gera uma ação de exploração e no segundo caso uma ação de exploração. O fator de exploração ϵ inicia com um valor alto (50%, por exemplo) e, conforme a simulação avança, diminui. Quando o valor de ϵ chegar a 0, o agente de seleção não irá mais explorar e passará a escolher sempre a ação com maior estimativa de recompensa.

8.3.4 Resultados

Os testes do desempenho do JOJTEAM foram realizados com a seguinte configuração:

- i)* o TeamBots foi utilizado como simulador dos robôs de pequeno porte e do jogo;
- ii)* o SMA funciona segundo o modelo MOISE^+ com uma organização formada por dois grupos, **team** e **ReorgGr**;
- iii)* a comunicação entre os agentes é feita com o SACI;
- iv)* o SMA do JOJTEAM é formado por:
 - 5 agentes com a arquitetura de jogadores,
 - 9 agentes como projetistas de reorganizações,
 - 1 agente monitor (que inicia 5 reorganizações),
 - 1 agente seletor (que usa o Q-Learning);
- v)* Os parâmetros de aprendizado são (estes valores foram ajustados empiricamente):
 - o valor de γ (desconto para recompensas futuras) é 0,9,
 - o valor inicial de ϵ (fator de exploração) é 0,5 e, a cada reorganização, diminui 0,0001 (em 1000 jogos o valor fator exploração chega a 0),
 - o valor de α (taxa de aprendizado) é inicialmente 0,2 e diminui em 0,00001 a cada reorganização;
- vi)* O time oponente é sempre o mesmo: o melhor time que acompanha a distribuição do TeamBots, chamado de DTeam.

Nesta configuração, os resultados alcançados estão resumidos no gráfico da figura 8.7. Para a construção do gráfico foram realizadas 40 seqüências de 2000 jogos. Considerando que p_{is} indica o placar final do i -ésimo jogo (eixo x no

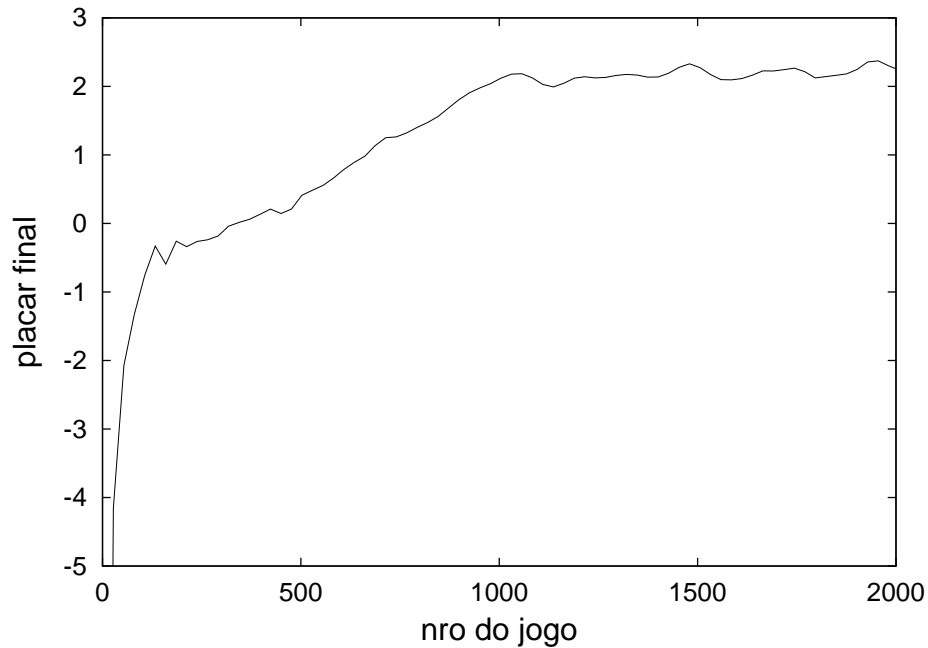


Figura 8.7: Resultados do aprendizado do time com estado do mundo incluindo tempo e placar.

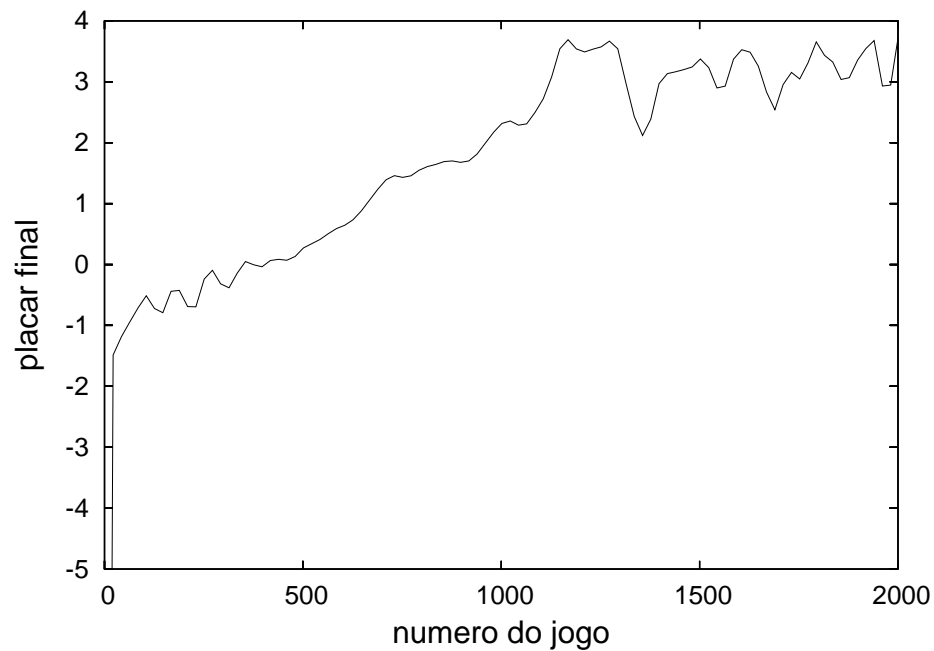
gráfico) da seqüência s , os valores do eixo y são dados pela média aritmética dos resultados dos i -ésimos jogos:

$$\frac{1}{40} \sum_{s=1}^{40} p_{ij}$$

Segundo estas simulações, o JOJTEAM passou a ganhar os jogos depois de ter jogado 480 partidas e, a partir do milésimo jogo, passou a ganhar com uma diferença de mais de dois gols. A política aprendida está representada na tabela 8.1 que foi obtida a partir da política mais comum aprendida nas 40 seqüências. Resumidamente, o time inicia com formação 4x1. No momento 1, se estiver perdendo por mais de dois gols, adota uma formação mais ofensiva (1x3x1), senão fica com o 4x1. No momento 2, volta para a formação 4x1 se estiver perdendo e não muda de formação se estiver ganhando. Contudo, se o time estiver ganhando por mais de 2 gols, aumenta a área dos jogadores. No momento 3, se estiver perdendo adota a formação mais ofensiva (1x1x3), se estiver ganhando fica com a formação 4x1. No momento 4, flexibiliza a área dos jogadores se estiver ganhando, caso contrário adota uma formação mais defensiva. Como era esperado, alguns projetistas ruins (como o Random) não tiveram suas propostas selecionadas.

Tabela 8.1: Política aprendida pelo JOJTEAM utilizando estado do mundo formado por tempo e placar

estado (tempo, placar)	ação
(0,-2)	begin
(0,-1)	begin
(0,0)	4x1
(0,1)	begin
(0,2)	begin
(1,-2)	1x3x1
(1,-1)	4x1
(1,0)	4x1
(1,1)	unflexGolie
(1,2)	nochange
(2,-2)	4x1
(2,-1)	4x1
(2,0)	nochange
(2,1)	nochange
(2,2)	flex
(3,-2)	1x1x3
(3,-1)	flexGolie
(3,0)	1x1x3
(3,1)	4x1
(3,2)	nochange
(4,-2)	4x1
(4,-1)	nochange
(4,0)	flex
(4,1)	flex
(4,2)	nochange



Política aprendida	
estado (tempo)	ação
0	4x1
1	newDS
2	4x1
3	1x3x1
4	flexGolie

Figura 8.8: Resultados do aprendizado do time com estado do mundo incluindo somente o tempo.

Além dos experimentos descritos acima, verificou-se também, nas mesmas condições, como o agente de seleção aprenderia caso a descrição dos estados do mundo (O) fosse formada unicamente pelo tempo. O placar do jogo não estaria mais representado nos estados, sendo utilizado apenas na função de recompensa. Os resultados estão ilustrados na figura 8.8. Como se pode perceber o time passou a ganhar um pouco antes do jogo 480 e obteve resultados melhores, pois o Q-Learning aprendeu mais rápido devido ao menor tamanho do conjunto Q . Sem o placar, o conjunto domínio de Q tem apenas 50 elementos.

8.4 Conclusões

A partir dos experimentos realizados neste segundo estudo do caso, pode-se enumerar as seguintes conclusões.

- i)* A viabilidade de usar reorganização mesmo em domínios que demandam reatividade dos agentes. Separar a equipe de jogadores da equipe de reorganização colaborou para que o JOJTEAM pudesse se reorganizar sem comprometer o desempenho dos jogadores.
- ii)* A viabilidade do uso de aprendizado por reforço na seleção. Esta técnica tem como principal vantagem a garantia de aproximação a uma política ótima de escolha de propostas.

Porém convém notar que o aprendizado nem sempre é possível, pois depende do domínio de aplicação poder ser modelado como um MDP e ter a possibilidade de ser simulado. Este último requisito não faz parte da teoria de aprendizado, mas no contexto de reorganização tem um caráter prático. Como o aprendizado por Q-Learning é um processo em geral lento, que demanda várias reorganizações e depende de parametrizações empíricas, um simulador do domínio é muito útil. Com o simulador, é possível aprender a política ótima antes do sistema ser colocado em produção.

- iii)* Da implementação do JOJTEAM e, particularmente, da sua configuração de reorganização, pôde-se constatar a simplicidade de se desenvolver projetistas de reorganizações (como ilustra a figura 8.6). Tal simplicidade é decorrente de três fatores:
 - a independência que o \mathcal{MOISE}^+ oferece entre as dimensões organizacionais (por exemplo, o Projetista newDS muda apenas as relações deonticas),
 - a noção de plano de reorganização que permite a mudança gradual da organização corrente, e
 - os projetistas não precisam definir bons planos de reorganização. Não precisam, por exemplo, utilizar um sofisticado sistema especialista para elaborar uma proposta de reorganização. Esta característica

deve-se à existência de uma etapa de seleção que faz a avaliação da qualidade ou da proposta ou do proponente.

Considerando o processo de reorganização como um problema de busca em um espaço infinito de novas organizações, os agentes projetistas fazem uma “filtragem” prévia neste espaço e o agente de seleção encontra uma nova organização neste sub-espaço.

9 Conclusões

Os problemas para um SMA se auto-reorganizar, enumerados na forma de questionamentos na introdução (página 8), foram, em parte, solucionados. As respostas propostas ao longo desta tese recaem sobre os problemas que puderam ser resolvidos por uma abordagem independente das arquiteturas específicas dos agentes, pois não se pretendeu restringir o grupo de agentes que participam do processo de reorganização. Por outro lado, definiu-se como um agente qualquer pode participar do processo.

9.1 Contribuições

Retomando as perguntas da introdução, as seguinte respostas destacam as principais contribuições da tese.

i) O que é uma organização?

A resposta a esta pergunta é feita por meio do modelo \mathcal{MOISE}^+ que define precisamente o que é uma organização e como ela contribui para que um SMA atinja sua finalidade. Resumidamente, uma organização é um conjunto de restrições comportamentais sobre os agentes, tais restrições objetivam tornar o sistema mais coeso e são descritas em três dimensões: estrutural, funcional e deôntica.

ii) Que propriedades um SMA deve possuir para facilitar seu processo de reorganização?

Por um lado, a simplificação do processo de reorganização é dada pela independência das dimensões organizacionais do SMA provida pelo \mathcal{MOISE}^+ . Por outro lado, o processo é simplificado pelo fato da organização necessária

para a execução da reorganização estar explicitamente descrita, o que foi chamado de a “organização do processo de reorganização”.

iii) Como identificar a necessidade de alterações organizacionais?

O modelo *MOISE*⁺ permite identificar uma falha organizacional de várias formas, por exemplo: a finalidade do sistema não estar sendo alcançada, má formação de grupos, baixa taxa de sucesso dos planos globais, etc. Porém, como se observou no estudo de caso do *JOJTEAM*, uma mudança organizacional pode ser motivada pela exploração de novas organizações com eventual incremento de desempenho.

iv) Quem deve realizar a reorganização?

A proposta apresentada é baseada na criação de um grupo especializado na condução da reorganização. Tal grupo é formado por agentes com habilidades de monitoração da organização, projeto de novas organizações e seleção destas propostas. A principal característica deste grupo é permitir a participação de vários agentes e portanto de várias técnicas de monitoração, projeto e seleção. Não são definidos quais são os agentes do grupo, mas como um agente participa do grupo.

v) Como projetar organizações para uma determinada finalidade?

Esta pergunta não teve uma resposta geral, apenas soluções dependentes do domínio de aplicação puderam ser formuladas.

vi) Como estimar o ganho de adotar uma organização em um dado ambiente?

As estratégias de seleção apresentadas, votação e aprendizado, estimam o ganho em se implementar uma nova organização baseadas em outras ocorrências de reorganização. O princípio utilizado pode ser intuitivamente descrito na frase “se anteriormente as propostas de um agente levaram a uma organização melhor, então sua proposta atual tem uma boa expectativa de ganho”.

vii) Como e por que agentes autônomos irão se submeter a uma (re)organização?

Esta pergunta não foi respondida propositadamente pois os aspectos individuais, como o funcionamento interno dos agentes, não eram objeto de estudo.

Adiciona-se ainda uma última pergunta:

viii) Por que utilizar este modelo de reorganização?

- Até onde foi realizada a revisão bibliográfica, esta proposta de reorganização é a única que lida com os problemas de seleção e implementação de um processo de reorganização.
- A linguagem de especificação organizacional, definida pelo MOISE^+ , permite descrever vários tipos e aspectos organizacionais. Por exemplo, é o único modelo que define cardinalidades e compatibilidades de papéis. Assim, o que é alterado pelo processo de reorganização pode ser bem detalhado.
- A reorganização ganhou um caráter de (meta) SMA, quem reorganiza são agentes e não procedimentos implícitos para os agentes. Por isso, é um sistema aberto e várias técnicas de seleção e projeto podem ser utilizadas.
- É um modelo que funcionou bem em dois estudos de caso e portanto mostrou-se viável.
- Possui uma implementação que simplifica sua utilização e garante que os agentes do SMA terão suas ações restringidas pela organização.

9.2 **Trabalhos futuros**

No decorrer do desenvolvimento da tese, foram feitas restrições de escopo para permitir o estudo pontual da questão de reorganização em um tipo particular de SMA. Além destas limitações, vários outros aspectos relevantes de investigação foram percebidos no decorrer do estudo e, por limitações de tempo e foco, não puderam ser desenvolvidos. Esta seção apresenta portanto os principais temas de continuação que podem ser desenvolvidos a partir da pesquisa realizada.

i) Quanto ao modelo organizacional:

- Complementar o modelo MOISE^+ com os itens do MOISE que não foram incluídos. Três elementos podem ser citados: (*i*) as ações que

os agentes devem realizar para satisfazer suas metas; (ii) os recursos que os agentes podem utilizar; e (iii) os protocolos de interação para a comunicação entre os agentes. Estes três elementos estão deonticamente ligados à EE e a EF. Por exemplo, um papel *pode* ou *deve* utilizar determinada ação ou recurso; um agente, na execução de uma missão em um esquema, *deve* seguir determinado protocolo. O estado do protocolo e do esquema são inter-dependentes, por exemplo, o envio de uma mensagem pode tornar uma meta permitida e a satisfação de uma meta pode permitir o envio/recebimento de uma mensagem.

ii) Quanto à arquitetura do modelo organizacional:

- Desenvolver uma arquitetura para agentes em sistemas organizados segundo o MOISE^+ . Como ponto de partida, poderia-se utilizar a arquitetura proposta por Hannoun (2002) baseada no ASIC (BOISSIER; DEMAZEAU, 1994). Um dos problemas que esta arquitetura deveria resolver é a conciliação entre as metas locais e globais.
- Juntamente com o item anterior, desenvolver uma arquitetura que desse suporte a sistemas do tipo AC (agentes com capacidade de representação organizacional em um sistema sem organização institucionalizada) propondo mecanismos que facilitassem a construção da organização observada.
- Estudar mecanismos de controle para que os agentes cumpram o que está estabelecido pela organização. No estudo de caso E-Alliance, por exemplo, foi feito uso da noção de penalidades. Este mecanismo, entretanto, deve ser melhor desenvolvido, principalmente nos aspectos de arquitetura de sistema.

iii) Quanto ao modelo de reorganização:

- A exemplo do trabalho de Carron e Boissier (2001), incluir marcações temporais nos elementos do MOISE^+ . Desta forma, o modelo de reorganizações contemplaria também as reorganizações pré-definidas. Por exemplo, o papel de lateral de um time existe somente no primeiro tempo do jogo.

-
- Avaliar como técnica de monitoração a diferença entre a organização institucionalizada de um sistema e sua organização observada. Uma diferença significativa sugere que a organização institucionalizada não está sendo adotada pelos agentes, o que pode caracterizar uma situação de falha organizacional.
 - Apesar de possível, não foram realizados experimentos onde o próprio grupo de reorganização é alvo do processo de reorganização.

Anexo A – SACI

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, algumas dificuldades são encontradas na sua utilização, tais como: *(i)* baixa interoperabilidade, pois geralmente pode-se utilizar uma única linguagem de programação ou 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) (HÜBNER; SICHMAN, 2000; ALBUQUERQUE et al., 2001). 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). 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 utilizando-se somente o nome do receptor, sendo sua localização na rede transparente;
- os agentes podem conhecer os outros por meio de um serviço de páginas amarelas;

- os agentes podem ser implementados como applets e terem sua interface em uma home-page;
- os agentes podem ser iniciados remotamente;
- possui ferramentas de monitoramento da atuação dos agentes que permitem a visualização e o armazenamento de eventos sociais, tais como entrada e saída de agentes no sistema, recebimento ou envio de mensagens.

A.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.

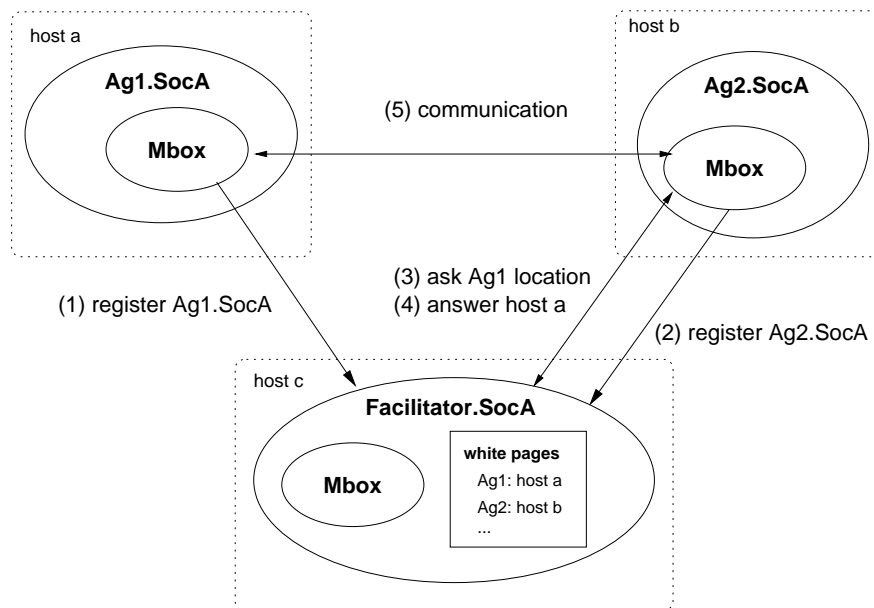
Quando um agente deseja entrar numa determinada sociedade, ele deve contatar o respectivo facilitador e registrar um nome. O facilitador irá verificar a unicidade deste nome e associá-lo à localização do agente. Da mesma forma, quando deseja sair da sociedade, deve também avisar o facilitador.

A.2 Envio e recebimento de mensagens

O ambiente SACI 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. A figura A.1 ilustra uma interação mais elaborada entre dois agentes.

A.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



O agente *Ag2* da sociedade SocA deseja se comunicar com o agente *Ag1* que pertence a 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 (seta (3)). O facilitador lhe responde “host a” (seta (4)). Recebida a localização, o *Ag2* inicia a comunicação com o agente *Ag1* (seta (5)).

Figura A.1: Exemplo de interação no SACI (HÜBNER; SICHMAN, 2000).

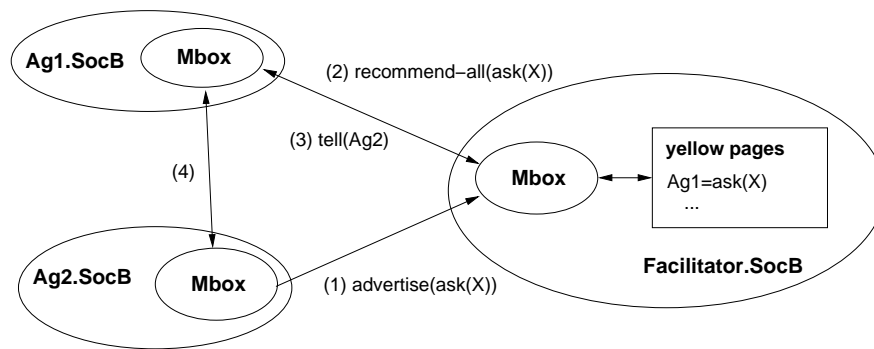


Figura A.2: Exemplo de anúncio de habilidades no SACI (HÜBNER; SICHMAN, 2000).

não conhece um agente capaz de realizá-lo, pode requisitar ao facilitador uma lista de agentes com tal habilidade. A figura A.2 ilustra um exemplo dessa interação.

A.4 Programação dos agentes com o SACI

O trecho de código mostrado a seguir, apresentado originalmente em (HÜBNER; SICHMAN, 2000), é um exemplo de utilização do ambiente SACI. 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();
        try {
            if (a.enterSoc("APlusServer")) {
                a.initAg(null);
            }
        } catch (Exception e) {
            System.err.println("Error="+e);
        }
    }
}
```

```

public void initAg(String[] args) {
    try {
        mbox.advertise("ask-one", "alg", "math", "X + Y");

        // add a message handler to answer sum asks
        mbox.addMessageHandler(null,"ask-one","alg","math",
            new MessageHandler() {
                public boolean processMessage(Message m) {
                    try {
                        Message r = new Message("tell");
                        r.put("receiver", m.get("sender"));
                        r.put("in-reply-to", m.get("reply-with"));
                        r.put("ontology", m.get("ontology"));
                        r.put("content", sum( (String)m.get("content") ));
                        mbox.sendMsg(r);
                    } catch (Exception e) {
                        System.err.println("Error sending message\n"+e);
                    }
                    return true;
                }
            });

    } catch (Exception e) {
        System.err.println("Error starting agent:"+e);
    }

    String sum(String formula) {
        // some code
    }
}

```

No método `main`, o agente entra na sociedade sugerindo uma identificação *APlusServer* e inicia sua execução. No método `initAg`, o agente anuncia suas habilidades (através do método `advertise`) e registra um receptor de mensagens indicando que o agente pode receber mensagens com a performativa `ask-one`, com

conteúdo escrito em qualquer linguagem, na ontologia `math` e com formato `X+Y`. Note que é através do objeto `mbox` que as funcionalidades do SACI são utilizadas. Por exemplo, no método `processMessage`, uma mensagem KQML é construída e enviada assincronamente pelo método `sendMsg`.

Um agente que solicita o serviço do agente desenvolvido acima poderia utilizar o seguinte método `run`:

```
public void run(String exp) {
    try {
        Vector receptores =
            mbox.consultYP("ask-one", null, "math", "X+Y");
        if (receptores.size() == 0) {
            System.out.println("No agent to answer me!");
        } else {
            for (int i=0; i < receptores.size(); i++) {
                String ag = (String) receptores.elementAt(i);
                Message m = new Message("(ask-one)");
                m.put("content", "\"" + exp + "\"");
                m.put("receiver", ag);
                Message r = mbox.ask(m);
                System.out.println(r.get("sender")+
                    " answers is " + r.get("content"));
            }
        }
    } catch (Exception e) { ... }
}
```

O método `consultYP` encapsula a consulta ao facilitador pelos agentes que têm a capacidade de somar. Para cada um destes agentes, é enviada uma mensagem solicitando a realização de uma soma. O método utilizado para esta solicitação (`ask(m)`) envia uma mensagem de performativa `ask` e espera a resposta para esta mensagem.

Anexo B – Exemplo de especificação organizacional no formato XML

Este anexo contém a especificação organizacional em formato XML (conforme utilizado na implementação do *MOISE+*) para os exemplos da escola (que foi desenvolvido no capítulo 3) e do time de futebol (desenvolvido no capítulo 8).

B.1 Exemplo da escola

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="xml/os.xsl" type="text/xsl" ?>
<!DOCTYPE OrganizationalSpecification SYSTEM "os.dtd">

<OrganizationalSpecification id="escola">
  <StructuralSpecification>
    <RolesDefinition>
      <Role id="docente"      <extends role="soc" /> </Role>
      <Role id="aluno"       <extends role="soc" /> </Role>
      <Role id="diretor"     <extends role="docente" /> </Role>
      <Role id="professor"  <extends role="docente" /> </Role>
    </RolesDefinition>
    <LinksType>
      <LinkType id="acquaintance" />
      <LinkType id="communication" />
      <LinkType id="authority" />
    </LinksType>

    <GroupSpecification id="escola">

      <Links>
        <Link source="docente"
              destination="docente"
              type="communication"
              scope="inter-group">
```

```
        extendsToSubGroups="true" />
</Links>

<SubGroups>

  <GroupSpecification id="corpodoc">
    <Roles>
      <Role id="professor" />
      <Role id="diretor" />
    </Roles>

    <Links>
      <Link source="diretor"
            destination="professor"
            type="authority"
            scope="intra-group"
            extendsToSubGroups="true" />
    </Links>

    <ConstrainFormation>
      <Compatibility source="professor"
                    destination="diretor"
                    scope="intra-group"
                    symmetric="false" />
      <Cardinality object="role"
                  id="diretor"
                  min="1" max="1" />
    </ConstrainFormation>
  </GroupSpecification>

  <GroupSpecification id="turma">
    <Roles>
      <Role id="professor" />
      <Role id="aluno" />
    </Roles>

    <Links>
      <Link source="docente"
            destination="aluno"
            type="authority"
            scope="intra-group"
            extendsToSubGroups="true" />
      <Link source="aluno"
            destination="professor"
            type="communication" />
    </Links>
  </GroupSpecification>
</SubGroups>
```

```

        scope="intra-group"
        extendsToSubGroups="true" />
    <Link source="aluno"
        destination="docente"
        type="acquaintance"
        scope="intra-group"
        extendsToSubGroups="true" />
    <Link source="aluno"
        destination="aluno"
        type="communication"
        scope="inter-group"
        extendsToSubGroups="true" />
</Links>

<ConstrainFormation>
    <Compatibility source="aluno"
        destination="professor"
        scope="inter-group"
        symmetric="true" />
    <Cardinality object="role"
        id="professor"
        min="1" max="1" />
    <Cardinality object="role"
        id="aluno"
        min="5" max="30" />
</ConstrainFormation>
</GroupSpecification>

</SubGroups>

<ConstrainFormation>
    <Cardinality object="role"
        id="aluno"
        min="0" max="300" />
    <Cardinality object="grSpec"
        id="corpodoc"
        min="1" max="1" />
</ConstrainFormation>
</GroupSpecification>

</StructuralSpecification>

<FunctionalSpecification>
    <Scheme id="prova1" rootGoal="realizarProva" >

        <Plan headGoal="realizarProva" operator="sequence">

```

```

        <Goal id="prepararProva" />
        <Goal id="reponderProva" />
    </Plan>

    <Mission id="m42" min="1" max="1">
        <Goal id="realizarProva" />
        <Goal id="prepararProva" />
    </Mission>

    <Mission id="m43">
        <Goal id="reponderProva" />
    </Mission>
</Scheme>

    <Preference mission="m43" preferable="m42" />
</FunctionalSpecification>

<DeonticSpecification>
    <DeonticOperatorDefinitions>
        <DeonticOperator id="obligation" />
        <DeonticOperator id="permission" />
    </DeonticOperatorDefinitions>

    <DeonticRelation type="obligation"
        role="professor"
        mission="m42"
        timeConstraint="Any" />

    <DeonticRelation type="obligation"
        role="aluno"
        mission="m43"
        timeConstraint="Any" />

</DeonticSpecification>
</OrganizationalSpecification>

```

B.2 Especificação do JOJTEAM

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="xml/os.xsl" type="text/xsl" ?>
<!DOCTYPE OrganizationalSpecification SYSTEM "os.dtd">

<OrganizationalSpecification id="joj-reorg-4">

```



```
<StructuralSpecification>
  <RolesDefinition>
    <Role id="player">      <extends role="soc" /> </Role>

    <Role id="back">
      <Property id="area" value="-1.37,0.76,0.2,-0.76" />
      <!-- area is a domain dependent information, the area
           the player must be -->
      <extends role="player" />
    </Role>

    <Role id="goalkeeper">
      <Property id="area" value="-1.37,0.5,-1.15,-0.5" />
      <extends role="back" />
    </Role>

    <Role id="left">
      <Property id="area" value="-1.37,0.76,1.37,0.0" />
      <extends role="player" />
    </Role>

    <Role id="right">
      <Property id="area" value="-1.37,0.0,1.37,-0.76" />
      <extends role="player" />
    </Role>

    <Role id="attacker">
      <Property id="area" value="-0.2,0.6,1.37,-0.6" />
      <extends role="player" />
    </Role>
  </RolesDefinition>

  <LinksType>
    <LinkType id="acquaintance" />
    <LinkType id="communication" />
    <LinkType id="authority" />
  </LinksType>

  <GroupSpecification id="team">
    <Property id="formation" value="2-2-1" />

    <Roles>
      <Role id="goalkeeper" />

      <Role id="back" />
    </Roles>
  </GroupSpecification>
</StructuralSpecification>
```

```
<Role id="left" />
<Role id="right" />
<Role id="attacker" />
</Roles>

<Links>
  <Link source="player" destination="player"
        type="communication"
        scope="inter-group" extendsToSubGroups="true" />
</Links>

<SubGroups>
  <LoadGroupSpecification uri="ReorgGr.xml" />
</SubGroups>

<ConstrainFormation>
  <Cardinality object="role" id="goalkeeper" min="1" max="1"/>
  <Cardinality object="role" id="back" min="1" max="1"/>
  <Cardinality object="role" id="left" min="1" max="1"/>
  <Cardinality object="role" id="right" min="1" max="1"/>

  <Cardinality object="role" id="attacker" min="1" max="1"/>

  <Cardinality object="grSpec" id="reorgGr" min="1" max="1"/>
</ConstrainFormation>
</GroupSpecification>

</StructuralSpecification>

<FunctionalSpecification>
  <Scheme id="playSoccer" rootGoal="scoreGoals" >

    <Plan headGoal="scoreGoals" operator="parallel">
      <Goal id="kickToGoal" />
      <Goal id="beAtOurGoal" />
      <Goal id="carryToGoal" />
    </Plan>

    <Mission id="mKG" min="4" max="4">
      <Goal id="kickToGoal" />
    </Mission>

    <Mission id="mBG" min="1" max="1">
      <Goal id="beAtOurGoal" />
    </Mission>

  </Scheme>
</FunctionalSpecification>
```

```
<Mission id="mCG" min="1" max="4">
  <Goal id="carryToGoal" />
</Mission>

</Scheme>
<Preference mission="mKG" preferable="mBG" />
</FunctionalSpecification>

<DeonticSpecification>
  <DeonticOperatorDefinitions>
    <DeonticOperator id="obligation" />
    <DeonticOperator id="permission" />
  </DeonticOperatorDefinitions>

  <DeonticRelation type="obligation" timeConstraint="Any"
    role="back" mission="mKG" />

  <DeonticRelation type="obligation" timeConstraint="Any"
    role="left" mission="mCG" />

  <DeonticRelation type="obligation" timeConstraint="Any"
    role="right" mission="mCG" />

  <DeonticRelation type="obligation" timeConstraint="Any"
    role="attacker" mission="mCG" />

  <DeonticRelation type="obligation" timeConstraint="Any"
    role="goalkeeper" mission="mBG" />
</DeonticSpecification>
</OrganizationalSpecification>
```

Referências Bibliográficas

- ALBUQUERQUE, R. L.; HÜBNER, J. F.; PAULA, G. E. de; SICHMAN, J. S.; RAMALHO, G. L. KSACI: A handheld device infrastructure for agents communication. In: INTERNATIONAL WORKSHOP AGENT THEORIES, ARCHITECTURES, AND LANGUAGES (ATAL-2001), 8., 2001, Seattle, WA. *Proceedings*. 2001. p. 327–337. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2001/Albuquerque-atal2001.pdf>>.
- ALVARES, L. O.; SICHMAN, J. S. Introdução aos sistemas multiagentes. In: MEDEIROS, C. M. B. (Ed.). *Jornada de Atualização em Informática (JAI'97)*. Brasília: UnB, 1997. cap. 1, p. 1–38.
- ANDREOLI, J. M.; CASTELLANI, S.; MUNIER, M. AllianceNet: Information sharing, negotiation and decision-making for distributed organizations. In: INTERNATIONAL CONFERENCE ON ELECTRONIC COMMERCE AND WEB (CECWEB'2000), 1., 2000, Greenwich. *Proceedings*. [S.l.], 2000.
- ARKIN, R. C. *Behavior-Based Robotics*. [S.l.]: MIT Press, 1998. (Intelligent Robotics and Autonomous Agents).
- BAEIJIS, C. *Fonctionnalité Emergente dans une Société d'Agents Autonomes – Etude des Aspects Organisationnels dans les Systèmes Multi-Agents Réactifs*. Tese (Thèse (Doctorat)) — LEIBNIZ-Institut IMAG, France, 1998.
- BALCH, T. *TeamBots*. 2000. Disponível em: <<http://www.teambots.org>>. Acesso em: 10 fev. 2003.
- BAZZAN, A. L. C.; LESSER, V. R.; XUAN, P. *Adapting an Organization Design through Domain-Independent Diagnosis*. Amherst, MA, 1994. Department of Computer Science/University of Massachusetts. Computer Science Technical Report, 1994-14.
- BOISSIER, O. *Problème du contrôle dans un système intégré de vision, utilisation d'un système multi-agents*. Tese (Thèse (Doctorat)) — INP Grenoble, 1993.
- BOISSIER, O. Modèles et architectures d'agents. In: BRIOT, J.-P.; DEMAIZEAU, Y. (Ed.). *Principes et architecture des systèmes multi-agents*. Paris: Hermes, 2002. cap. 2, p. 71–108.
- BOISSIER, O.; DEMAIZEAU, Y. ASIC: An architecture for social and individual control and its applications to computer vision. In: EUROPEAN WORKSHOP ON MODELING AUTONOMOUS AGENTS IN MULTI-AGENT WORLD (MAAMAW'94), 6., 1994, Odense, Denmark. *Proceedings*. [S.l.], 1994.

- BORDINI, R. H. *Suporte Lingüístico para Migração de Agentes*. 128 p. Dissertação (Dissertação de Mestrado) — Instituto de Informática/UFRGS, Porto Alegre, 1994.
- BORDINI, R. H. *Contributions to an Anthropological Approach to the Cultural Adaptation of Migrant Agents*. 238 p. Tese (These (Ph.D.)) — University College London, London, 1999.
- BORDINI, R. H.; VIEIRA, R.; MOREIRA, Á. F. Fundamentos de sistemas multiagentes. In: FERREIRA, C. E. (Ed.). *Jornada de Atualização em Informática (JAI'01)*. Fortaleza, Brasil: SBC, 2001. v. 2, cap. 1, p. 3–44.
- BREZILLON, P.; PASQUIER, L.; POMEROL, J. Representing knowledge by contextual graphs. In: IBERO-AMERICAN CONFERENCE ON AI/BRAZILIAN SYMPOSIUM ON AI (IBERAMIA/SBIA'2000), 7/15., 2000, Atibaia, São Paulo, Brazil. *Proceedings (Paper Track)*. Berlin: Springer, 2000. (LNAI 1952), p. 245–258.
- BRIOT, J.-P.; DEMAZEAU, Y. (Ed.). *Principes et architecture des systèmes multi-agents*. Paris: Hermes, 2002.
- CAIRE, G.; COULIER, W.; GARIJO, F.; GOMEZ, J.; PAVON, J.; LEAL, F.; CHAINHO, P.; KEARNEY, P.; STARK, J.; EVANS, R.; MASSONET, P. Agent oriented analysis using MESSAGE/UML. In: INTERNATIONAL WORKSHOP ON AGENT-ORIENTED SOFTWARE ENGINEERING (AOSE-2001), 2., 2001, Montreal, Canada. *Proceedings*. [S.l.], 2001.
- CARRON, T.; BOISSIER, O. Towards a temporal organizational structure language for dynamic multi-agent systems. In: MODELING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'2001), 10., 2001, Annecy. *Pre-Proceeginds*. [S.l.], 2001.
- CASTELFRANCHI, C. Social power: A point missed in multi-agent, DAI and HCI. In: DEMAZEAU, Y.; MÜLLER, J.-P. (Ed.). *Decentralized Artificial Intelligence*. Amsterdam: Elsevier, 1990.
- CASTELFRANCHI, C. Commitments: From individual intentions to groups and organizations. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS'96), 2., 1996, Kyoto, Japan. *Proceedings*. [S.l.]: AAAI Press, 1996. p. 41–48.
- CASTELFRANCHI, C.; CONTE, R.; PAOLUCCI, M. Normative reputation and the cost of compliance. *Journal of Artificial Societies and Social Simulation*, v. 1, n. 3, 1998. Disponível em: <<http://jasss.soc.surrey.ac.uk/1/3/3.html>>. Acesso em: 20 dez. 2002.
- CAVEDON, L.; SONENBERG, L. On social commintment, roles and preferred goals. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS'98), 3., 1998, Paris, France. *Proceedings*. [S.l.]: IEEE Press, 1998. p. 80–87.

- COHEN, P. R.; LEVESQUE, H. J. Intention = choice + commitment. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., 1987, Seattle, WA. *Proceedings*. [S.l.]: Morgan Kaufmann, 1987. p. 410–415.
- COHEN, P. R.; LEVESQUE, H. J. Confirmation and joint action. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI'91), 12., 1991, Sydney, Australia. *Proceedings*. [S.l.], 1991. p. 951–959.
- CONTE, R.; CASTELFRANCHI, C. Minds and norms: Types of normative reasoning. In: . [S.l.]: Cambridge University Press, 1994.
- CONTE, R.; CASTELFRANCHI, C. *Cognitive and Social Action*. [S.l.]: UCL Press, 1995.
- CONTE, R.; FALCONE, R.; SARTOR, G. Agents and norms: How to fill the gap? *Artificial Intelligence and Law*, v. 7, p. 1–15, 1999.
- CONTE, R.; GILBERT, N.; SICHMAN, J. S. MAS and social simulation: A suitable commitment. In: MULTI-AGENT SYSTEMS AND AGENT-BASED SIMULATION, 1., 1998, Paris, France. *Proceedings*. Berlin: Springer, 1998. (Lecture Notes in Artificial Intelligence 1534), p. 1–9.
- COSTA, A. C. da R.; HÜBNER, J. F.; BORDINI, R. H. On entering an open society. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 11., 1994, Fortaleza. *Anais*. Fortaleza, 1994. p. 535–546. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/1994/sbia/Rocha-sbia94.ps>>. Acesso em: 20 fev. 2003.
- DECKER, K.; LESSER, V. Task environment centered design of organizations. In: AAAI SPRING SYMPOSIUM ON COMPUTATIONAL ORGANIZATION DESIGN, 1994, Stanford. *Proceedings*. [S.l.], 1994.
- DECKER, K. S. TÆMS: A framework for environment centered analysis and design of coordination mechanisms. In: O'HARE, G. M. P.; JENNINGS, N. R. (Ed.). *Foundations of Distributed Artificial Intelligence*. New York: John Wiley, 1996. cap. 16, p. 429–447.
- DECKER, K. S. Task environment centered simulation. In: PRIETULA, M. J.; CARLEY, K. M.; GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Menlo Park: AAAI Press / MIT Press, 1998. cap. 6, p. 105–128.
- DECKER, K. S.; LESSER, V. *Designing a Family of Coordination Algorithms*. [S.l.], 1995. UMass Computer Science Technical Report 1994-14.
- DECKER, K. S.; LESSER, V. R. Generalizing the partial global planning algorithm. *International Journal of Intelligent Cooperative Information Systems*, v. 1, n. 2, p. 319–346, 1992.

- DEMAZEAU, Y.; MÜLLER, J.-P. (Ed.). *Decentralized Artificial Intelligence*. Amsterdam: Elsevier, 1990.
- DIGNUM, V.; DIGNUM, F. Modelling agent societies: Co-ordination frameworks and institutions. In: PORTUGUESE CONFERENCE ON ARTIFICIAL INTELLIGENCE (EPIA'01), 10., 2001, Porto, Portugal. *Proceedings*. Berlin: Springer, 2001. p. 191–204. LNAI 2258.
- DROGOUL, A.; CORBARA, B.; LALANDE, S. MANTA: New experimental results on the emergence of (artificial) ant societies. In: GILBERT, N.; CONTE, R. (Ed.). *Artificial Societies: the Computer Simulation of Social Life*. London: UCL Press, 1995. p. 119–221.
- DURFEE, E. H.; LESSER, V. R. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 21, n. 5, p. 1167–1183, sep/oct 1991.
- DURFEE, E. H.; MONTGOMERY, T. A. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 21, n. 6, p. 1363–1378, november/december 1991.
- ESTEVA, M.; RODRIGUEZ-AGUIAR, J. A.; SIERRA, C.; GARCIA, P.; ARCOS, J. L. On the formal specification of electronic institutions. In: AGENT-MEDIATED ELECTRONIC COMMERCE, 2001. *Proceedings*. Berlin: Springer, 2001. (LNAI 1191), p. 126–147.
- FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. London: Addison-Wesley, 1999.
- FERBER, J.; GUTKNECHT, O. A meta-model for the analysis and design of organizations in multi-agents systems. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS'98), 3., 1998, Paris, France. *Proceedings*. [S.l.]: IEEE Press, 1998. p. 128–135.
- FININ, T.; FRITZON, R.; MCKAY, D.; MCENTIRE, R. KQML: A language and protocol for knowledge and information exchange. In: INTERNATIONAL WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE, 13., 1994, Seattle, WA. *Proceedings*. [S.l.], 1994. p. 126–136.
- FOX, M. S. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 11, n. 1, p. 70–80, Jan 1981.
- FOX, M. S.; BARBUCEANU, M.; GRUNINGER, M.; LON, J. An organizational ontology for enterprise modeling. In: PRIETULA, M. J.; CARLEY, K. M.; GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Menlo Park: AAAI Press / MIT Press, 1998. cap. 7, p. 131–152.
- FRANKLIN, S.; GRAESSER, A. Is it an agent or just a program? A taxonomy for autonomous agents. In: INTERNATIONAL WORKSHOP ON AGENT

THEORIES, ARCHITECTURES, AND LANGUAGES, 3., August 12-13, 1996, Budapest, Hungary. *Proceedings*. Berlin: Springer, 1997. (Lecture Notes in Computer Science, Vol. 1193), p. 21–35. ISSN 3-540-62507-0.

GALBRAITH, J. *Organization Design*. [S.l.]: Addison-Wesley, 1977.

GARIJO, F.; GÓMES-SANZ, J. J.; PAVÓN, J.; MASSONET, P. Multi-agent system organization: An engineering perspective. In: MODELING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'2001), 10., 2001, Annecy. *Pre-Proceedings*. [S.l.], 2001.

GASSER, L. Organizations in multi-agent systems. In: MODELING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'2001), 10., 2001, Annecy. *Proceedings*. Annecy, 2001.

GLASER, N.; MORIGNOT, P. The reorganization of societies of autonomous agents. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'97), 8., 1997, Ronneby, Sweden. *Proceedings*. Berlin: Springer, 1997. (LNAI 1237), p. 98–111.

GROSZ, B. J.; KRAUS, S. Collaborative plans for complex group action. *Artificial Intelligence*, v. 86, p. 269–357, 1996.

HANNOUN, M. *MOISE: un modèle organisationnel pour les systèmes multi-agents*. Tese (Thèse (Doctorat)) — École Nationale Supérieure des Mines de Saint-Etienne, 2002.

HANNOUN, M.; BOISSIER, O.; SICHMAN, J. S.; SAYETTAT, C. MOISE: An organizational model for multi-agent systems. In: IBERO-AMERICAN CONFERENCE ON AI/BRAZILIAN SYMPOSIUM ON AI (IBERAMIA/SBIA'2000), 7/15., 2000, Atibaia, São Paulo, Brazil. *Proceedings (Paper Track)*. Berlin: Springer, 2000. (LNAI 1952), p. 152–161.

HIROAKI, K.; ASADA, M.; KUNIYOSHI, Y.; NODO, I.; OSAWA, E. The robot world cup initiative. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENT (AGENTS-97), 1., 1997, Marina del Rey, CA, USA. *Proceedings*. [S.l.]: ACM Press, 1997.

HORLING, B.; BENYO, B.; LESSER, V. Using self-diagnosis to adapt organizational structures. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTES (AGENTS' 01), 5., 2001, Montreal, Canada. *Proceedings*. [S.l.], 2001.

HORLING, B.; VINCENT, V. L. R.; BAZZAN, A.; XUAN, P. Diagnosis as an integral part of multi-agent adaptability. In: DARPA INFORMATION SURVIVABILITY CONFERENCE AND EXPOSITION, 2000, South Carolina. *Proceedings*. [S.l.]: IEEE Computer Society, 2000. p. 211–219.

HÜBNER, J. F. *Migração de Agentes em Sistemas Multi-Agentes Abertos*. Dissertação (Dissertação de Mestrado) — CPGCC da UFRGS, Porto Alegre,

1995. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/dissertacao/dissertacao.pdf>>. Acesso em: 20 fev. 2003.

HÜBNER, J. F.; SICHMAN, J. S. SACI: Uma ferramenta para implementação e monitoração da comunicação entre agentes. In: INTERNATIONAL JOINT CONFERENCE, 7TH IBERO-AMERICAN CONFERENCE ON AI, 15TH BRAZILIAN SYMPOSIUM ON AI (IBERAMIA/SBIA 2000), 7/15., 2000, Atibaia, São Paulo, Brazil. *Proceedings (Open Discussion Track)*. São Carlos: ICMC/USP, 2000. p. 47–56. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2000/Hubner-iberamia2000.pdf>>. Acesso em: 20 fev. 2003.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. MOISE⁺: Towards a structural, functional, and deontic model for MAS organization. In: FIRST INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS (AAMAS'2002), 1., 2002, Bologna, Italy. *Proceedings*. ACM Press, 2002. p. 501–502. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2002/Hubner-aamas2002.pdf>>. Acesso em: 20 fev. 2003.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE (SBIA'02), 16., 2002, Porto de Galinhas, PE, Brazil. *Proceedings*. Berlin: Springer, 2002. (LNAI 2507), p. 118–128. ISSN 0302-9743. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2002/Hubner-sbia2002.pdf>>. Acesso em: 20 fev. 2003.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. Spécification structurelle, fonctionnelle et déontique d'organisations dans les systèmes multi-agents. In: JOURNÉES FRANCOPHONES INTELLIGENCE ARTIFICIELLE DISTRIBUÉE & SYSTÈMES MULTI-AGENTS (JFIADSMA'02), 10., 2002, Lille, France. *Actes*. Lille, France: Hermès, 2002. p. 205–216. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2002/Hubner-jfiadsma2002.pdf>>. Acesso em: 20 fev. 2003.

IGLESIAS, C.; GARRIJO, M.; GONZALEZ, J. A survey of agent-oriented methodologies. In: . [S.l.]: "Springer-Verlag: Heidelberg, 1999. p. 317–330.

ISHIDA, T.; GASSER, L.; YOKOO, M. Organization self-design of distributed production systems. *IEEE Transactions on Knowledge and Data Engineering*, v. 4, n. 2, p. 123–134, abril 1992.

JENNINGS, N. R.; WOOLDRIDGE, M. J. *Agent Technology: foundations, applications, and markets*. London: Springer Verlag, 1998.

JOHNSON, S. *Emergence: the connected lives of ants, brains, cities, and software*. New York: Touchstone, 2001.

- KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Reserach*, v. 4, p. 237–285, 1996.
- KINNY, D.; LJUNGBERG, M.; RAO, A.; SONENBERG, E.; TIDHAR, G.; WERNER, E. Planned team activity. In: MODELING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD (MAAMAW'92), 4., July 1992, Martino el Cimino, Italy. *Proceedings*. Berlin: Springer, 1994. p. 227–256. (LNAI 830).
- LEMAÎTRE, C.; EXCELENTE, C. B. Multi-agent organization approach. In: IBEROAMERICAN WORKSHOP ON DAI AND MAS, 2., 1998, Toledo, Spain. *Proceedings*. [S.l.], 1998.
- LEVESQUE, H. J.; COHEN, P. R.; NUNES, J. H. T. On acting together. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI-90), 8., 1990, Boston. *Proceedings*. Menlo Park: AAAI Press / MIT Press, 1990. p. 94–99.
- LUGO, G. G.; HÜBNER, J. F.; SICHTMAN, J. S. Representação e evolução de esquemas sociais em sistemas multi-agentes: Um enfoque funcional. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 3., 2001, Fortaleza, CE, Brazil. *Anais*. Fortaleza, Brazil: SBC, 2001. p. 1237–1246. Disponível em: <<http://www.inf.furb.br/~jomi/pubs/2001/Lugo-enia2001.pdf>>. Acesso em: 20 fev. 2003.
- MALONE, T. W. Tools for inventing organizations: Toward a handbook of organizational process. *Management Science*, v. 45, n. 3, p. 425–443, March 1999.
- MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill, 1997.
- MURPHY, R. R. *Introduction to AI robotics*. [S.l.]: Bradford Book/MIT Press, 2000.
- ODELL, J.; PARUNAK, H. V. D.; BAUER, B. Extending UML for agents. In: . [S.l.: s.n.], 2000.
- PATTISON, H. E.; CORKILL, D. D.; LESSER, V. R. Instantiating description of organizational structures. In: HUHNS, M. N. (Ed.). *Distributed Artificial Intelligence*. London: Morgan Kaufmann, 1987. v. 1, cap. 3, p. 59–96.
- PRASAD, M. N.; DECKER, K.; GARVEY, A.; LESSER, V. Exploring organizational design with TÆMS: A case study of distributed data processing. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS'96), 2., 1996, Kyoto, Japan. *Proceedings*. [S.l.]: AAAI Press, 1996. p. 283–290.
- PRIETULA, M. L.; CARLEY, K. M.; GASSER, L. A computational approach to organiations and organizing. In: PRIETULA, M. J.; CARLEY, K. M.;

- GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Menlo Park: AAAI Press / MIT Press, 1998. cap. 1, p. xiii–xix.
- RAO, A. S.; GEORGEFF, M. P. BDI agents: from theory to practice. In: INTERNATIONAL CONFERENCE ON MULTIAGENT SYSTEMS (ICMAS'95), 1., 1995, San Francisco, USA. *Proceedings*. [S.l.]: AAAI Press, 1995. p. 312–319.
- ROMESÍN, H. M.; GARCÍA, F. J. V. *De máquinas e seres vivos, Autopoiese: A organização do vivo*. 3. ed. Porto Alegre: Artes Médicas, 1997. ISBN 85-7307-302-0.
- RUSSEL, S.; NORVIG, P. *Artificial Intelligence: a modern approach*. New Jersey: Prentice-Hall, 1995.
- SEARLE, J. R. *Intentionality: An essay in the philosophy of mind*. [S.l.]: Cambridge, 1983.
- SICHMAN, J. S. *Du Raisonment Social Chez les Agents: Une Approche Fondée sur la Théorie de la Dépendance*. Tese (Thèse (Doctorat)) — Institut National Polytechnique de Grenoble, 1995.
- SMITH, R. G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transaction on Computers*, v. 29, n. 12, p. 1104–1113, 1980.
- SO, Y.; DURFEE, E. H. An organizational self-design model for organizational change. In: AAAI93 WORKSHOP ON AI AND THEORIES OF GROUPS AND ORGANIZATIONS, 1993. *Proceedings*. [S.l.], 1993.
- SO, Y.; DURFEE, E. H. Designing tree-structure organizations for computational agents. *Computational & Mathematical Organization Theory*, Kluwer Academic, v. 2:3, 1996.
- STEELS, L. Cooperation between distributed agents through self-organization. In: DEMAZEAU, Y.; MÜLLER, J.-P. (Ed.). *Decentralized Artificial Intelligence*. Amsterdam: Elsevier, 1990. p. 175–196.
- STONE, P.; VELOSO, M. M. Task decomposition and dynamic role assignment for real-time strategic teamwork. In: INTERNATIONAL WORKSHOP AGENT THEORIES, ARCHITECTURES, AND LANGUAGES (ATAL-98), 5., 1998, Paris. *Proceedings*. Berlin: Springer, 1999. (LNCS 1555), p. 293–308.
- STRUGEON, E. L. *Une Méthodologie d'Auto-Adaptation d'un Système Multi-Agents Cognitifs*. Tese (Thèse (Doctorat)) — Iniversié de Valenciennes et du Haut-Cambrésis, Valenciennes, France, 1995.
- SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Cambridge: Bradford, 1998.

- TAMBE, M. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, v. 7, p. 83–124, 1997.
- TAMBE, M.; PYNADATH, D. V.; CHAUVAT, N. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, v. 4, n. 2, 2001.
- TERABE, M.; WASHIO, A.; KATAI, O.; SAWARAGI, T. A study of organizational learning in multiagents systems. In: WEISS, G. (Ed.). *Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-Agent Environment*. [S.l.]: Springer, 1997, (LNAI 1221). p. 168–179.
- WEISS, G. *Some studies in distributed machine learning and organizational design*. [S.l.], 1994. Disponível em: <citeseer.nj.nec.com/weiss94some.html>.
- WEISS, G. (Ed.). *Multiagent Systems: A modern approach to distributed artificial intelligence*. London: MIT Press, 1999. 619 p. ISBN 0-262-23203-0.
- WERNER, E. Cooperating agents: A unified theory of communication and social structure. In: GASSER, L.; HUHNS, M. N. (Ed.). *Distributed Artificial Intelligence*. Pitman, London: Morgan Kaufmann, 1989. v. 2, cap. 1, p. 3–36.
- WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. [S.l.]: John Wiley and Sons, 2002.
- WOOLDRIDGE, M.; JENNINGS, N. R.; KINNY david. A methodology for agent-oriented analysis and design. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTES (AGENT'S 99), 3., 1999, Seattle, WA. *Proceedings*. [S.l.]: ACM, 1999.

Índice Remissivo

- AALAADIN, 21
- agente
- autônomo, 2
 - cognitivo, 5
 - entrada e saída, 97
 - reativo, 5
- AgentUML, 31
- agMis*, 72
- agRole*, 72
- aprendizado por reforço, 142
- ASIC, 100
- auto-organização, 117
- autonomia de planejamento, 103
- autonomia organizacional, 19
- campos de potência, 181
- cardinalidade, 47
- compatibilidade, *veja* papel
- comportamento, 37
- configuração de reorganização, 127
- coordenação, 32, 60, 107
- E-Alliance, 157
- entidade organizacional, 71
- dinâmica, 86
- especificação
- deôntica, 68
 - estrutural, 48
 - funcional, 65
 - organizacional, 38
 - dinâmica, 148
- esquema motor, 181
- esquema social, 57, 61
- dinâmica, 92–94
- eventos organizacionais, 86
- eventos reorganizacionais, 148
- exploração, 146
- exploração, 146
- falha organizacional, 117
- finalidade, 13, 66
- first*, 87
- futebol de robôs, 176
- GAIA, 29
- getHead*, 61
- getObligatedMissions*, 102
- getPossibleGoals*, 103
- getRootGoal*, 87
- grPlayers*, 87
- grType*, 72
- grupo
- dinâmica, 89
 - especificação, 45
- gState*, 72
- herança, *veja* papel
- isCommitted*, 95
- isLeafGoal*, 62
- isPermitted*, 95
- isPossible*, 95
- isRootGr*, 48
- isSatisfied*, 57
- JoJTeam, 177, 210
- ligação, *veja* papel
- MDP, 143
- MESSAGE, 31
- meta
- folha, 60
- metas, 56
- dinâmica, 94–96
 - pré-condições, 60, 95
 - sub, 59
 - super, 59

- missão, 27, 57
 comprometimento, 99
 preferências, 64
mo, 62
modelo de contrato, 168
modelo de reorganização, 122
MOISE, 27, 78
MOISE⁺, 36–82

ng, 45
nm, 62
np, 45

obrigações organizacionais, 102
organização, 17
 falha, 117
 institucionalizada, 15
 observada, 14
 tipos, 15
organização objetiva, *veja* organização institucionalizada
OrgBox, 86
OrgManager, 85

papel, 21, 23, 27, 81
 abstrato, 40
 adoção, 98
 compatibilidades, 43
 herança, 23, 41
 ligações, 28, 42
 social, 41
plano, 27, 59
preferências, *veja* missão
proposta de mudança, 132

Q-Learning, 145

relações deônticas, 27, 68
reorganização, 112
 agente, 116
 endógena, 116
 exógena, 117
 exemplos, 114
 formas, 114
 objeto, 112
restrição forte, 18
restrição fraca, 18
RoboCup, *veja* futebol de robôs

SACI, 85, 201
schPlayers, 87
scType, 72
second, 87
sistemas abertos, 3, 109
sistemas multiagentes, 1
sub-metas, 59
subGr, 72
subgroups, 87
super-metas, 59
superGr, 87

TÆMS, 24
tarefa, 24
TeamBots, 177
teoria organizacional, 7
TOVE, 22

visão centrada na organização, 14
visão centrada nos agentes, 14