Marcos Augusto Hochuli Shmeil

Sistemas Multiagente na Modelação da Estrutura e Relações de Contratação de Organizações

Faculdade de Engenharia
Universidade do Porto
1999

Marcos Augusto Hochuli Shmeil

Sistemas Multiagente na Modelação da Estrutura e Relações de Contratação de Organizações

Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto para a obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores, realizada sob a orientação científica do Doutor Eugénio da Costa Oliveira, Professor Associado com Agregação da Faculdade de Engenharia da Universidade do Porto

catalogada

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto
1999

007978

681.3(043) SHMsm/SIS
M
UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA
N.º 42459
CDU
Data / / 19

À Josélia, à Anneluize, à Angélica, à Lizianne e
aos meus pais
com todo o carinho

Agradecimentos

Nem sempre há oportunidade de agradecermos a todas as pessoas, as quais participaram em nossas vidas contribuindo para a nossa formação, seja de forma directa ou indirecta. A escrita de uma tese apresenta esta oportunidade. Gostaria primeiramente de agradecer ao meu orientador Professor Doutor Eugénio da Costa Oliveira, não só pela sua orientação científica, bem como pelos seus ensinamentos sobre investigação e ciência. Sua postura crítica, sensata, franca e por vezes dura, mas rigorosa e consistente perante o quotidiano, muito me influenciou. Obrigado Doutor Eugénio.

O ambiente no qual vivemos possui a característica de facilitar ou dificultar o alcance de objectivos. O NIA&R (Núcleo de Inteligência Artificial e Robótica) caracterizou-se como um moto de BEM ESTAR. Quero registar os meus agradecimentos ao Alcínio ("*in memoriam*"), à Ana Paula, à Benedita, à Conceição, ao Isidro, ao José Manuel, ao Luiz Paulo, ao Mouta, ao Nuno, ao Paulo Lopes, ao Paulo Marinho, ao Rui Camacho e ao Tozé. Sempre atentos às minhas preocupações nunca se furtaram em deixar disponível uma ideia, uma palavra, enfim uma ajuda na ânsia de minimizá-las. Obrigado NIA&R.

Complexas de serem mensuradas, porém mais fáceis de serem sentidas, assim são as relações familiares. Um especial agradecimento à minha esposa Josélia e às minhas meninas, por terem tido a paciência e terem mais uma vez acreditado no trabalho do esposo e pai. Gostaria que vocês soubessem do real valor de estarem comigo neste empreendimento. Obrigado por vocês existirem em minha vida.

Mesmo na ausência de meu pai Harold ("*in memoriam*"), quero que a minha mãe Ermelinda saiba da importância que é tê-los como pais. Obrigado.

Sem os componentes, recursos financeiros e tempo, é impossível concretizar certos projectos. Quero agradecer ao CNPq - Conselho Nacional de Pesquisa - Brasil, à PUC-Pr - Pontifícia Universidade Católica do Paraná - Brasil e ao BANESTADO-

Banco do Estado do Paraná S.A. - Brasil. Sem haverem tornado disponíveis os recursos citados, não teria havido sonho concretizado.

Ao país de acolhimento e suas entidades, mais especificamente a Faculdade de Engenharia da Universidade do Porto através do NIA&R do Departamento de Engenharia Electrotécnica e de Computadores meus agradecimentos pelos recursos computacionais e metodológicos cedidos.

Por fim aos meus inúmeros parentes e amigos com os quais partilhei os meus sonhos, medos e ansiedades, e dos quais sempre obtive apoios concretos, obrigado.

Resumo

Organizações, como qualquer entidade inerentemente complexa e distribuída, são caracterizadas pelas suas interacções internas e externas. Geralmente, e como resultado de um processo interactivo e contínuo, as organizações envolvidas, tornam-se mais eficientes/eficazes. Esta atitude, alcançada através da optimização dos recursos, pode ser vista como resultado do "saber fazer" adquirido nessas experiências.

Em termos gerais, o trabalho desta dissertação pode ser classificado como uma contribuição para o estudo e modelação computacional do comportamento de organizações. Em particular, o interesse recai numa relação interorganizacional específica: o processo de selecção que conduz ao estabelecimento de contratos entre organizações. O processo de selecção pode ser caracterizado como um laço interactivo composto de uma fase de validação, seguida de uma fase de negociação. Durante a actividade de selecção, conflitos podem ocorrer, sendo a negociação utilizada como o seu meio de resolução. De acordo com as diferentes metodologias de selecção que possam ser adoptadas, diferentes oportunidades de aprendizagem podem ser detectadas.

O modelo desenvolvido (bem como o correspondente sistema informático) que suporta o processo interactivo, acima mencionado, é chamado de "*ARTOR*" (*ARTificial ORganizations*), e é baseado em paradigmas da Inteligência Artificial Distribuída (IAD), mais especificamente o de Sistemas Multiagente (SMA) e Aprendizagem Simbólica Automática (AAS). As organizações modeladas e que compõem a comunidade, bem como os componentes dessas organizações são representadas por agentes computacionais providos de capacidades para desempenharem os seus papeis sociais.

Para além dos princípios da autoridade e da subdivisão de tarefas baseadas em critérios (similaridade de métodos e de recursos utilizados, compartilhamento do tempo, interdependência de acções) que conduzem a uma divisão intra-organizacional (gerando uma estrutura para a organização), a caracterização das actividades organizacionais em dois papeis básicos também se apresenta. Esses papeis são o de coordenação e de execução de actividades. O papel da coordenação procura assegurar a execução das actividades planeadas e o papel da execução procura a própria

realização das actividades planeadas. Estes papeis sendo executados por diferentes agentes são úteis, não só no suporte para a autoridade e para a subdivisão de tarefas, mas também para o processo de avaliação do planeamento, uma vez que decresce a relevância do conhecimento individual (interesses pessoais) e acresce a relevância do conhecimento global (interesse organizacional).

O sistema "*ARTOR*" pode ser visto como uma metodologia de modelação de organizações no que respeita a algumas das suas funcionalidades. Mais especificamente foram desenvolvidos processos de negociação sucessiva entre agentes dotados de intenções e objectivos autónomos, conducentes ao estabelecimento de contractos tidos como benéficos para todas as intervenientes. Protocolos específicos e métodos de relaxação de restrições foram usados na aproximação desenvolvida.

Tentou-se ainda dotar os agentes desenvolvidos com capacidades de evolução de acordo com a análise das interacções realizadas.

A intenção do trabalho realizado é pois a de contribuir para o desenvolvimento de métodos que permitam a modelação progressiva de organizações interagindo num mercado cada vez mais assente na comunicação electrónica.

Abstract

Organizations, as any complex and inherently distributed entity, are characterized by their internal and external interactions. In general, and as a result of the continuous mutual interactive process, the involved organizations become more and more efficient/efficacious. This new attitude, achieved through optimization of resources, can be seen as the outcome of a know-how acquired from previous interactions.

In broad terms, the work presented in this dissertation can be classified as a contribution to the study and the computational modeling of the behavior of organizations. In particular, it is concerned with a specific inter-organization relationship: the selection process that leads to the establishment of contracts between organizations. This selection process can be characterized as an iterative loop composed of an evaluation phase followed by a negotiation phase. During the selection activity, conflicts may occur imposing the negotiation as a means for conflict resolution. According to the diversity of selection methodologies that can be adopted, also different learning opportunities can be detected.

The developed approach (as well as the corresponding system based on computational resources), which supports the interaction process mentioned above, is called "*ARTOR*" (*ARTificial ORganizations*), and is based on Distributed Artificial Intelligence - Multi-Agent Systems (DAI-MAS) and Machine Learning (ML) paradigms. The organizations in the community, as well as the components of these organizations are represented as computational agents, which are provided with the capabilities to play their social roles.

The authority, and task decomposition based on different criteria (similarities of methods and resources, time sharing and interdependence of actions) which lead to an intra-organizational division (establishing a structure for the organization). The organization activities are also characterized in two basic roles. These roles are the coordination and the realization of activities. The role of coordinating activities tries to guarantee the possibility of executing the planned activities and the role of executing activities takes care of the real execution of the planned activities. These roles being played by different agents are useful, not only to support the authority and the task decomposition, but also to the evaluation of the performance of each one,

once that they decrease the importance of the individual knowledge (personal concern), and increase the importance of the global knowledge (organizational concern).

The "*ARTOR*" system can be seen as a methodology to model the organizations concerning some of their functionalities. Specifically a successive negotiation process has been developed among agents possessing autonomous intentions and goals leading to the establishing of contracts which grant benefits to all counterparts. Specific protocols and methods of constraint relaxation have been used in the development of this approach. In addition, we have tried to endow the agents with evolution capabilities according to the interactions carried out.

Our work has the purpose of giving a contribution to the development of methods that, in a progressive way model organizations interacting into a market more and more based on electronic communications.

Résumé

Les organisations, comme n'importe quelle entité inséparablement complexe et distribuée, sont caractérisées par leurs interactions internes et externes. En général, et comme résultat d'un processus interactif et continu, les organisations concernées deviennent plus efficientes/efficaces. Cette attitude, atteinte par le moyen de l'optimisation des ressources, peut être vue comme résultat du "savoir-faire" acquis dans ces expériences.

En termes généraux, le travail de cette dissertation peut être classé comme un apport à l'étude et modelisation informatique du comportement des organisations. En particulier, l'intérêt retombe sur une relation interorganisationnelle spécifique: le processus de sélection qui mène à l'établissement de contrats parmi des organisations. Le processus de sélection peut être caractérisé comme un lien interactif composé d'une phase de validation, suivie d'une phase de négociation. Pendant l'activité de sélection, des conflits peuvent avoir lieu, la négociation étant utilisée comme leur moyen de résolution. En accord avec les différentes méthodologies de sélection qui puissent être adoptées, de différentes opportunités d'apprentissage peuvent être décelées.

Le modèle développé (ainsi que le système informatique correspondant) qui supporte le processus interactif susmentionné est appelé "*ARTOR*" (*ARTificial Organizations),* et est fondé sur des paradigmes de l'Intelligence Artificiel Distribuée (IAD), plus spécifiquement celui des Systèmes Multi-agent (SMA) et Apprentissage Symbolique Automatique (ASA). Les organisations modelées et qui composent la communauté, ainsi que les composants de ces organisations sont représentées par des agentes informatiques pourvus des aptitudes pour tenir leurs rôles sociaux.

Au-delà des principes de l'autorité et de la subdivision des tâches fondées sur des critères (similarité de méthodes et de ressources utilisées, partage du temps, interdépendance d'actions) qui mènent à une division intra-organisationnelle (engendrant une structure pour l'organisation), la caractérisation des activités organisationnelles en deux rôles de base se présente aussi. Ces rôles sont celui de la coordination et de l'exécution d'activités. Le rôle de la coordination cherche à assurer l'exécution des activités planifiées et le rôle de l'exécution cherche la propre réalisation des activités planifiées. Ces rôles étant exécutés par des agents différents

sont utiles, non seulement au support de l'autorité et de la subdivision des tâches, mais aussi au processus d'évaluation du plan, vu que décroît l'importance de la connaissance individuelle (intérêts personnels) et grandit l'importance de la connaissance globale (intérêt organisationnel).

Le système "*ARTOR*" peut être vu comme une méthodologie de modelisation d'organisations en ce qui concerne quelques-unes de leurs fonctionnalités. Plus spécifiquement ont été développés des processus de négociation successive entre agents doués d'intentions et d'objectifs autonomes menant à l'établissement de contrats considérés comme bénéfiques pour toutes les participantes. Des protocoles spécifiques et des méthodes de relâchement de restrictions ont été employés dans l'approche développée.

On a essayé encore de douer les agents développés avec des capacités d'évolution en accord avec l'analyse des interactions réalisées.

L'intention du travail réalisé est donc celle de contribuer au développement de méthodes permettant la modelisation progressive d'organisations interagissant dans un marché de plus en plus assis sur la communication électronique.

Índice

Lista de Figuras XIV

Capítulo 1 INTRODUÇÃO
1.1 Motivação 01
1.2 Enquadramento 03
1.3 Inteligência Artificial 05
1.3.1 Inteligência Artificial Distribuída 06
1.3.1.1 Sistemas Multiagente 08
1.3.1.1.1 Definições 08
1.3.1.1.2 Classificações 12
1.3.1.1.3 Implementações 14
1.3.1.1.4 Interacções 16
1.3.1.1.5 Arquitecturas 19
1.3.1.1.6 Linguagens 20
1.3.2 Aprendizagem 21
1.3.2.1 Um Pouco de História sobre a Obtenção e Origem do Conhecimento 21
1.3.2.2 Aprendizagem Automática 23
1.3.2.2.1 Classificações 25
1.3.2.3 Aprendizagem em Sistemas Multiagente 29
1.3.2.3.1 Arquitecturas 31
1.3.2.3.2 Conclusão 33
1.4 A Organização da Tese 34
1.5 Principais Contributos e Potenciais Áreas de aplicação 35
1.5.1 Introdução 35
1.5.2 Contributos 36
1.5.3 Aplicações 37
1.6 Conclusão 37

Capítulo 2 A MODELAÇÃO DE UMA SOCIEDADE DE ORGANIZAÇÕES
2.1 Introdução 40
2.2 As Organizações 41
2.2.1 Os Componentes do Ambiente Interno ("*inner*") das Organizações 41
2.2.2 Arquitecturas do Ambiente Interno ("*inner*") das Organizações 46
2.2.2.1 Modelo de "*Skarmeas*" 46
2.2.2.2 Modelo de *Barbuceanu e Fox*" 48
2.2.2.3 Modelo "*ADEPT*" 49
2.2.2.4 Modelo "*OSD*" 50
2.3 Um Cenário para uma Sociedade de Organizações 52
2.3.1 Tipos de "Ambientes" 52
2.3.2 Mercado Descentralizado 54
2.3.3 Exemplo de Mercado Descentralizado 54
2.3.4 Modelos Baseados na Descentralização 55
2.3.5 Mercado Centralizado 56
2.3.6 Modelo "*Walras*" 56
2.3.7 Conclusão 57
2.4 Organizações Virtuais 58
2.4.1 Definições 58
2.4.2 Cooperação em Estruturas Virtuais 58
2.4.3 O Projecto "*AVE*" 59
2.4.4 O Modelo de "*Rabelo* e *Camarinha-Matos*" 60
2.5 Organizações Artificiais - Sistema "*ARTOR*" 62
2.5.1 Arquitectura do Sistema "*ARTOR*" 62
2.5.2 Definição do Sistema "*ARTOR*" 63
2.5.3 Os Componentes do Sistema "*ARTOR*" - As Organizações 72
2.5.3.1 Perspectiva Conceptual Organizacional para um Agente Envolvente 72
2.5.3.2 Perspectiva Computacional para um Agente Envolvente 81
2.5.3.2.1 A Comunicação 88
2.5.3.2.2 A Base de Conhecimento 91
2.5.3.2.3 A Coordenação 95
2.5.4 Os Componentes do Sistema "*ARTOR*" - O Quiosque "*News_Stand*" 105
2.6 Conclusões 109

Capítulo 3 A RELAÇÃO SOCIAL, INTERORGANIZAÇÕES, DE SELECÇÃO E DE CONTRATAÇÃO DE COLABORADORES PARA O SISTEMA "*ARTOR*"
3.1 Introdução 112
3.2 Modelos para Suporte ao Processo de Negociação 114
3.3 Trabalhos Relativos a Automatização do Processo de Negociação 118
3.3.1 Modelo de "*Bazaar*" 118

3.3.2 Negociação para "*Rosenchein* e *Zlotkin*" 119
3.3.3 Negociação para "*J.P.Müller*" 119
3.3.4 Sequenciamento da Produção como um Processo de Negociação 121
3.4 O Processo de Selecção e de Contratação de Colaboradores em "*ARTOR*" 124
3.4.1 O convite à Sociedade de Organizações 131
3.4.2 A avaliação das Propostas e a Situação de Conflito 134
3.4.3 A Metodologia de Negociação para a Resolução de Conflitos 139
3.4.4 A Contratação 142
3.4.5 Aprendendo com o Processo de Negociação 144
3.5 Conclusões 150

Capítulo 4 CONCLUSÕES 151
4.1 Limitações 155
4.2 Trabalhos Futuros 155

REFERÊNCIAS 157

ANEXO A Sistemas "*BDI*" ("*Beliefs, Desires and Intentions*") 166
ANEXO B Linguagens de Comunicação Intra e Interagentes 170
ANEXO C Propriedades Gerais de Agentes e as Propriedades Específicas 180
para Agentes Deliberativos
ANEXO D Arquitecturas de Agentes 183
ANEXO E Linguagens para Programação de Agentes 188
ANEXO F Metodologias de Aprendizagem Simbólica Automática: 190
Indução Empírica e Dedução Axiomática
ANEXO G Sistema Multiagente com a Capacidade de Aprendizagem 197
ANEXO H Metodologia "*ARIS*" 204

Lista de Figuras

Capítulo 1

[fig. 1] O mundo real
[fig. 2] A relação social de contratação
[fig. 3] Contributos das disciplinas
[fig. 4] As quatro categorias de enquadramento da IA
[fig. 5] O ambiente distribuído
[fig. 6] Interacção entre o computador-cliente e o computador-servidor
[fig. 7] Os "*mobile agents*"
[fig. 8] Algumas das dimensões para classificação de agentes
[fig. 9] Os três tipos de estruturas numa comunidade de agentes
[fig. 10] O "*UIA*" numa comunidade cooperativa de agentes
[fig. 11] A composição de uma mensagem
[fig. 12] As duas arquitecturas básicas para a modelação de agentes
[fig. 13] Sequência de acontecimentos a partir do estímulo até a formação de um conceito
[fig. 14] Classificação dos métodos de aprendizagem automática
[fig. 15] Organização em busca de parceiros
[fig. 16] Modelo genérico da capacidade de aprendizagem para um agente
[fig. 17] Estrutura da dissertação

Capítulo 2

[fig. 18] A divisão das actividades organizacionais
[fig. 19] Os canais formais e informais nas organizações
[fig. 20] Os três mecanismos de coordenação nas organizações
[fig. 21] As estruturas organizacionais
[fig. 22] A estrutura de um escritório
[fig. 23] A arquitectura do agente no modelo de "*Skarmeas*"
[fig. 24] O agente de informação no modelo de "*Barbuceanu* e *Fox*"
[fig. 25] As agências do modelo "*ADEPT*"

[fig. 26] Uma sociedade de organizações
[fig. 27] Tipos de mercado
[fig. 28] Arquitectura de um agente em um ambiente descentralizado
[fig. 29] O processo de ofertas no sistema "*WALRAS*"
[fig. 30] Estruturas virtuais criada a partir da decomposição de uma actividade global
[fig. 32] - A Sociedade modelada pelo sistema "*ARTOR*"
[fig. 33] O sistema "*ARTOR*" ao nível mais abstracto
[fig. 34] O descritor global do sistema "*ARTOR*"
[fig. 35] O descritor do mundo
[fig. 36] O descritor dos modelos
[fig. 37] O descritor dos artefactos
[fig. 38] O descritor do modelo "*artor*"
[fig. 39] O descritor do artefacto "*artor*"
[fig. 40] O lançador da sociedade no ambiente "*ARTOR*"
[fig. 41] Ecrã geral do sistema "*ARTOR*"
[fig. 42] Explicação sobre o conhecimento contido no sistema "*ARTOR*"
[fig. 43] Prosseguimento no lançamento da sociedade
[fig. 44] Organização, agentes e estrutura
[fig. 45] Características das actividades nas dimensões e nas estruturas de um agente envolvente
[fig. 46] O agente envolvente de menor granularidade
[fig. 47] Estruturas organizacionais modeladas através de agentes envolventes
[fig. 48] Estruturas virtuais
[fig. 49] O conhecimento corporativo em um agente envolvente
[fig. 50] Janela para as mensagens recebidas
[fig. 51] Arquitectura dos agentes
[fig. 52] A capacidade de comunicação
[fig. 53] Janela para as mensagens recebidas
[fig. 54] Estrutura do planeamento global
[fig. 55] Capacidades da base de conhecimento corporativa
[fig. 56] Espaço de comportamentos de um agente gestor para o exercício da capacidade de coordenação
[fig. 57] Agenda de um agente
[fig. 58] Explicação sobre a estrutura de um agente envolvente
[fig. 59] Arquitectura do quiosque "*News-Stand*"
[fig. 60] O sistema "*ARTOR*"

Capítulo 3

[fig. 61] Matriz de retribuição
[fig. 62] Jogo da rivalidade
[fig. 63] Matriz para o Dilema dos Prisioneiros
[fig. 64] Comportamentos para a relação social de selecção e de contratação de colaboradores
[fig. 65] Taxinomia dos domínios de avaliação
[fig. 66] Regiões dos valores dos domínios dos critérios

[fig. 67] As utilidades, para um agente, dos valores do domínio de uma "*SC*"
[fig. 68] Espaço de possibilidades da "*CL*", de "*artor_org_A*", de provimento de P_1
[fig. 69] Espaço de possibilidades da "*CL*" , de "*artor_org_1*", de provimento de P_1
[fig. 70] Espaço de possibilidades da "*CL*" , de "*artor_org_2*", de provimento de P_1
[fig. 71] Comparação dos valores dos critérios ofertados e contra-ofertados
[fig. 72] Oferta inicial gerada pelo organizador
[fig. 73] Contra-oferta inicial gerada pelos respondentes
[fig. 74] Uma sessão de negociação
[fig. 75a] Resultados das ofertas e contra-ofertas da sessão de negociação - 1ª parte
[fig. 75b] Resultados das ofertas e contra-ofertas da sessão de negociação -2ª parte
[fig. 76b] Resultado obtido pelo organizador, referente ao processo de Selecção e de Contratação de Colaboradores - 1ª parte
[fig. 76b] Resultado obtido pelo organizador, referente ao processo de Selecção e de Contratação de Colaboradores -2ª parte
[fig. 77a] Estados das utilidades das ofertas e das contra-ofertas - 1ª parte
[fig. 77b] Estados das utilidades das ofertas e das contra-ofertas - 1ª parte

Capítulo 4

[fig. 78] O sistema "*ARTOR*"

Anexo A

[fig. A1] Arquitectura de um agente "*BDI*"

Anexo B

[fig. A2] Arquitectura de um agente gerado pelo "*UPShell*"
[fig. A3] "*Interfaces*" entre os nós do sistema multiagente
[fig. A4] Primeiro caso de protocolos
[fig. A5] Segundo caso de protocolos
[fig. A6] Terceiro caso de protocolos

Anexo C

[fig. A7] As propriedades de um agente deliberativo

Anexo G

[fig. A8] A decomposição de um problema
[fig. A9] Os principais componentes do modelo proposto por Sian
[fig. A10] O quadro de interacções do modelo proposto
[fig. A11] Classificação das condições climática e produtos
[fig. A12] O quadro de interacções do domínio do comércio de mercadorias

Anexo H

[fig. A13] O processamento de uma ordem de produção/fornecimento
[fig. A14] As visões de um modelo de processos
[fig. A15] Os quatro elementos que compõem o "*ARIS*"
[fig. A16] O processo de ordem de produção/ fornecimento
[fig. A17] Árvore da função "tratamento da ordem de produção/ fornecimento"

Capítulo 1

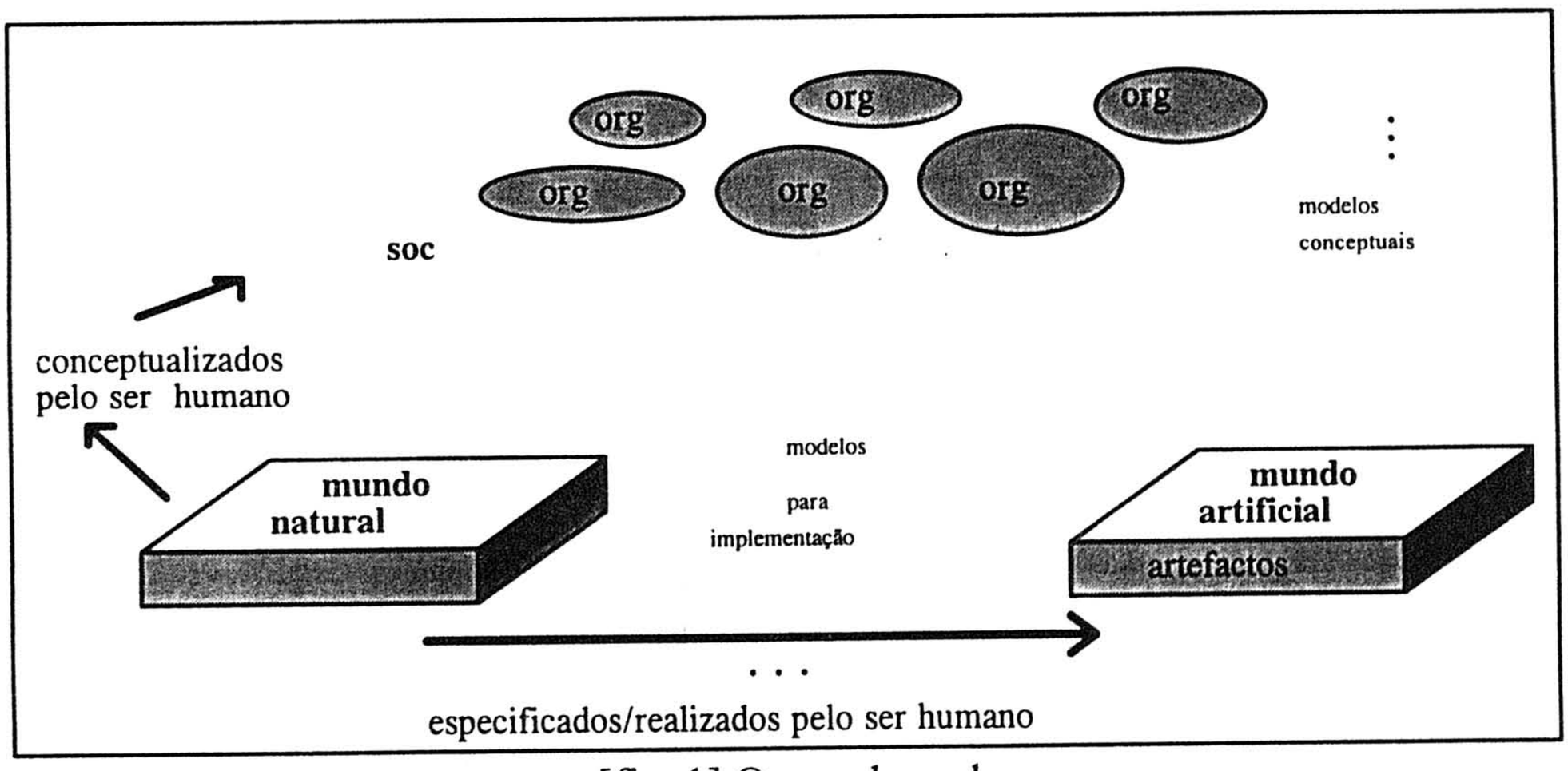
INTRODUÇÃO

1.1 Motivação

Componente e protagonista de um mundo natural (objectos e fenómenos estudados pelas ciências naturais) e criador de um mundo artificial (artefactos *man-made* [1]), o ser humano, durante o seu período de vida, está constantemente modificando o estado do mundo real (natural e artificial).

Estas modificações de estados são orientadas por **modelos** [fig. 1] e ocorrem através de comportamentos (conjunto de unidades de transformação) baseados nas suas capacidades cognitivas e motoras [2], resultado de três diferentes componentes:

i) percepção,
ii) memória, e
iii) raciocínio.



[fig. 1] O mundo real

O ser humano procura com estes novos estados do mundo real, obter a satisfação das suas necessidades (sobrevivência, conforto, reconhecimento, etc.), das

suas intenções e dos seus desejos. Porém depara-se com limitações cognitivas e motoras (acções) para obtenção de maneira individual de muitas dessas satisfações. Como um dos meios para minimizar os efeitos dessas limitações, modela o mundo real em **sociedades** (classes ou agrupamentos de classes de objectos do mundo real e suas relações). Estrutura estas sociedades através de organizações humanas (doravante denominadas apenas de **organizações**), nas quais cada indivíduo disponibiliza as suas capacidades (cognitivas e/ou motoras), apresentando um comportamento definido como inteligente. Este comportamento inteligente é caracterizado (em graus diferentes) pelas habilidades de perceber o ambiente que o rodeia (subconjunto do mundo real), tomar decisões, estabelecer relações, coordenar acções, reconhecer objectos e eventos, gerar planos e aprender. Os objectivos que até então eram gerados pelas necessidades, intenções e desejos individuais, são então acrescidos por objectivos organizacionais e sociais, cujo componente básico para a realização são as relações (**n:n**) estabelecidas entre seres humanos, organizações e sociedades. "A sociedade moderna e industrializada é uma sociedade de organizações, das quais cada ser humano passa a depender para nascer, viver e morrer [3] ".

As organizações numa sociedade de organizações, a exemplo de qualquer outra entidade inerentemente **complexa** [4] e **distribuída** [5], podem ser caracterizadas pelas suas interacções intra e interorganizações. Em geral, como resultado desse contínuo processo de interacções, as organizações envolvidas incrementam e refinam seus conhecimentos, permitindo que se tornem mais eficientes e eficazes. A melhoria nos seus desempenhos, alcançada através da optimização e normalização dos recursos envolvidos, pode ser compreendida como resultado do "*know-how*" adquirido nas interacções.

Então, motivado:

i) em geral, pela modelação de organizações que desempenham os papeis de **consumidoras**, **fornecedoras** e **produtoras** num mercado de bens e serviços, e pelas relações intra e intersociais dessas organizações,

ii) em particular pela relação social de **contratação**, suportada pelo processo de selecção de colaboradores para o fornecimento de bens e serviços, o qual é um dos principais componentes da acção para a terceirização das organizações ("*outsourcing*"), bem como o da criação de **empresas virtuais**,

iii) pela necessidade de processos automatizados e inteligentes de **tomada de decisão** dotados de um subconjunto das características que são encontradas no comportamento de peritos humanos no domínio,

iv) pela necessidade de preservar e usufruir do **conhecimento** envolvido nos processos de tomada de decisão, raciocinando e **aprendendo** com tais experiências,

v) pelo potencial que o domínio da Inteligência Artificial Distribuída apresenta na modelação de problemas complexos e **distribuídos**, bem como o de aplicação em problemas do mundo real, e

vi) pela atenção que a comunidade científica tem dispensada [67, 69, 78, 88, 89, 90, 91] no domínio da modelação das organizações e das

relações intra e intersociais através de metodologias de Inteligência Artificial,

este trabalho de tese, propõe e discute uma metodologia para a construção de artefactos computacionais no domínio da modelação de organizações com ênfase na relação social de contratação de organizações para o fornecimento de um bem ou serviço.

1.2 Enquadramento

Para modelar organizações e relações sociais intra e interorganizações através de componentes passíveis de tratamento por computador, há a necessidade de recorrermos a dois grupos de disciplinas;

(i) as que auxiliam na compreensão do problema a ser estudado e modelado, e

(ii) as que contribuem com conceitos e metodologias, que são suportadas através de recursos computacionais e permitem a implementação da solução adoptada.

No primeiro grupo, ou seja, as que auxiliam na compreensão do problema, fundamentalmente encontra-se: a Teoria das Organizações, a Teoria Geral da Administração, a Economia na sua vertente Microeconómica, a Sociologia, e a Psicologia. De entre as que fazem parte do segundo grupo, destaca-se a Inteligência Artificial, mais especificamente a Inteligência Artificial Distribuída no seu ramo de Sistemas Multiagente e a Aprendizagem Simbólica Automática.

Com a finalidade de enquadrar o tema da tese nas disciplinas acima referidas, será primeiramente necessário enunciar o problema a ser estudado e modelado. Para tal, consideremos o seguinte cenário:

"Dada uma sociedade de organizações, na qual:

(i) existem organizações, orientadas por seus planos, que procuram o alcance de objectivos,

(ii) existe um subconjunto de organizações que para o alcance de seus objectivos, estão interessadas em contratar organizações ou competências destas que cooperem no alcance dos mesmos, através do fornecimento de bens ou serviços,

(iii) a escolha de colaboradores ocorre através de um processo de selecção, baseado em critérios de avaliação, e

(iv) exista um subconjunto de organizações que possuem a intenção de cooperar,

então, baseadas nos critérios de avaliação, as organizações interessadas em contratar outras, promovem um processo de **selecção:**

(i) deixando disponível um convite à sociedade, convidando as organizações a participar de um processo de selecção,

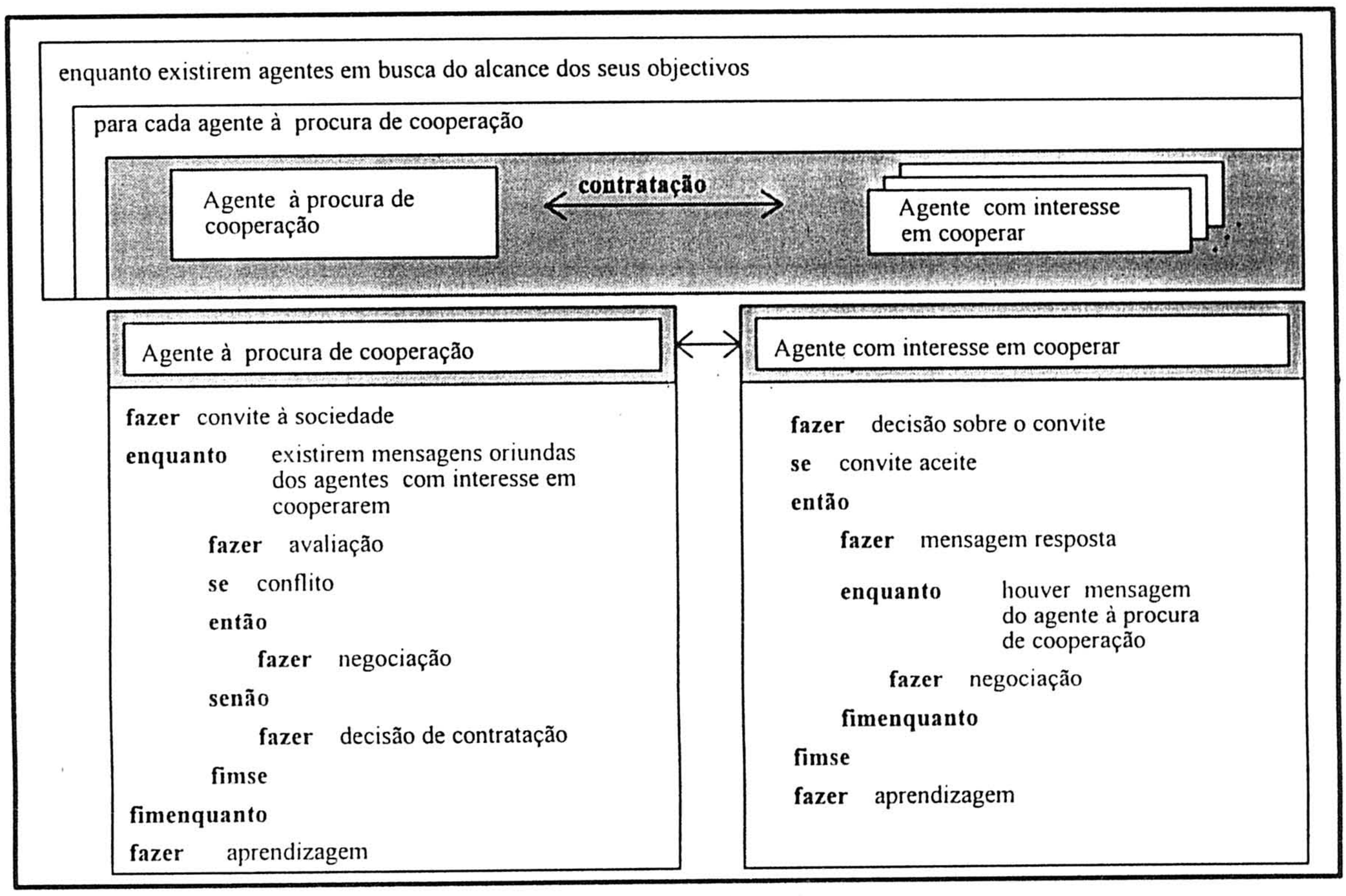
(ii) recebendo e avaliando as respostas dadas pelas organizações interessadas sobre o convite formulado,

(iii) negociando se necessário, num certo número de interacções, uma vez apresentada uma situação de conflito, e

(iv) contratando uma organização, caso exista, que satisfaça os critérios de avaliação em valores convenientes em cada um de seus domínios.

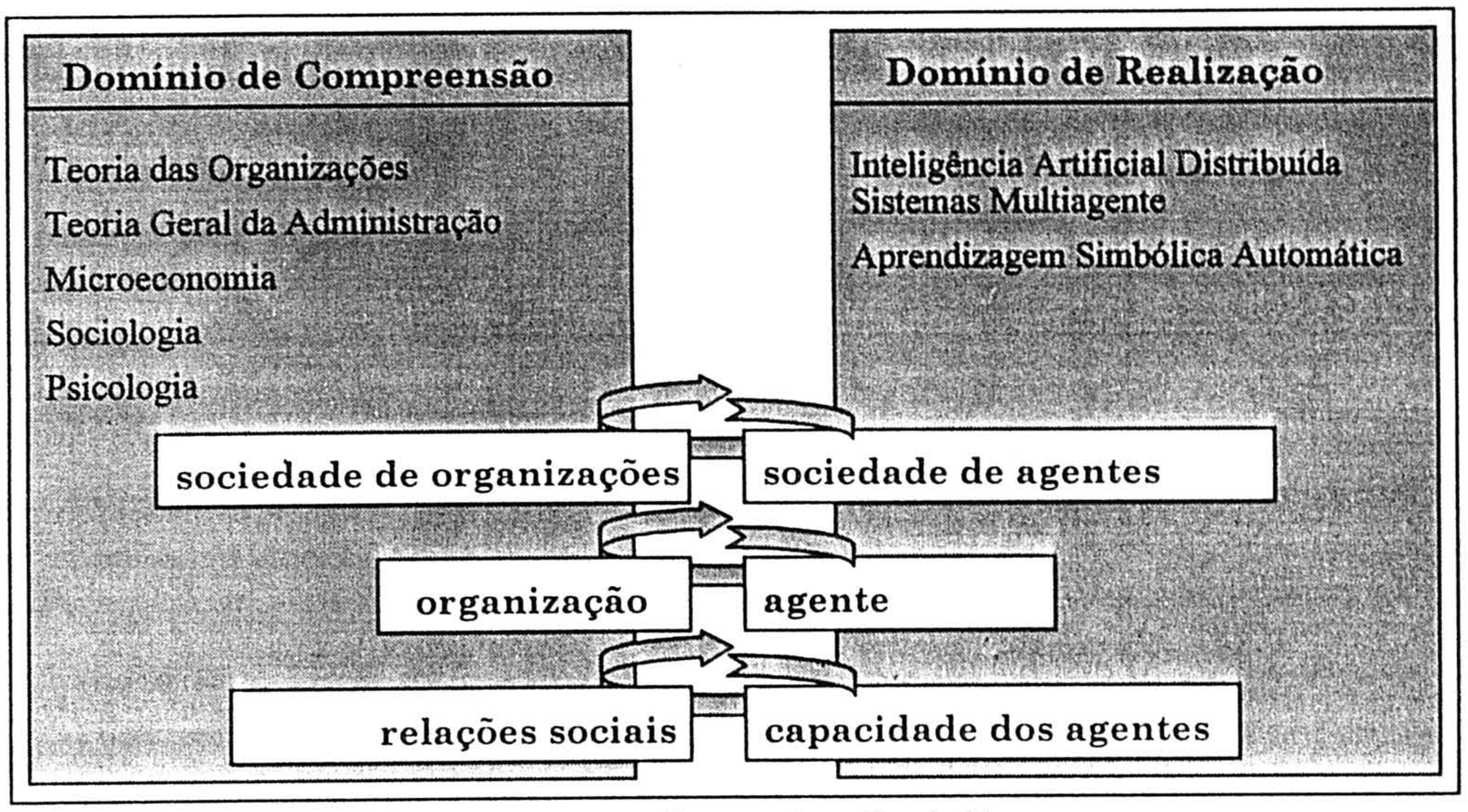
Por outro lado, as organizações interessadas em cooperar, tendo como objectivo serem as contratadas, e orientadas pelos seus interesses, procuram atender as restrições impostas em cada um dos critérios de avaliação. Baseadas nas experiências proporcionadas pela relação social de contratação, as organizações envolvidas potenciam a possibilidade de aprenderem conceitos sobre o comportamento dos colaboradores, no domínio da relação".

Tendo em consideração o cenário acima descrito, o tema central da tese coloca-se na elaboração de uma aproximação baseada em Inteligência Artificial, que modele as organizações da sociedade de organizações, como agentes computacionais (artefactos) e a relação social de contratação como uma das capacidades desses agentes. A [fig. 2] apresenta a relação social de contratação numa sociedade de agentes organizacionais.



[fig. 2] A relação social de contratação

Quanto à fundamentação do presente trabalho, e como citado anteriormente, as disciplinas apresentadas na [fig. 3], foram as que de forma relevante contribuíram a nível de compreensão e implementação da solução adoptada.



[fig. 3] Contributos das disciplinas

Devido ao tema proposto se apresentar como sendo multidisciplinar nos vectores compreensão, modelação e implementação, leva-nos a representar as conexões entre os elementos do cenário e as disciplinas utilizadas através de uma hélice circular. Procura-se com esta representação explicar que, se por um lado recorremos às disciplinas como a Teoria das Organizações, Teoria Geral da Administração, etc, para a compreensão dos elementos do cenário, por outro, além de recorrermos às disciplinas da Ciência da Computação para a modelação e a implementação através de recursos computacionais, importamos conhecimento do domínio de compreensão. Esta acção ocorre num contínuo vaivém, contribuindo para a expansão do conhecimento sobre o elemento em estudo, nas várias disciplinas que o suportam.

1.3 Inteligência Artificial

A Inteligência Artificial (IA) busca, por um lado, a compreensão, e por outro, o desenvolvimento de entidades inteligentes. Tem como objectivo maior, uma teoria sobre inteligência, que explique o comportamento que naturalmente ocorre em entidades inteligentes. Ela orienta a criação de artefactos (entidades artificiais), capazes de exibir comportamento inteligente. Em [6] encontramos algumas definições de IA oriundas de diferentes autores, cujas variações são enquadradas ao longo de duas dimensões, originando quatro categorias [fig. 4].

Sistemas que pensam como os humanos	Sistemas que pensam racionalmente
Sistemas que agem como os humanos	Sistemas que agem racionalmente

[fig. 4] As quatro categorias de enquadramento da IA

Da comparação interlinhas ressaltam as definições cujo ênfase está nos processos de pensamento/raciocínio ou nos de comportamento/acção. Da análise

intercolunas destacam-se as definições cujas características incluem as do modo humano de execução ou as baseadas num conceito ideal de inteligência denominado de racionalidade.

Ainda com relação às aproximações utilizadas para a construção de sistemas artificiais que exibam características presentes em entidades ditas inteligentes, dois cursos de acção são considerados [7]:

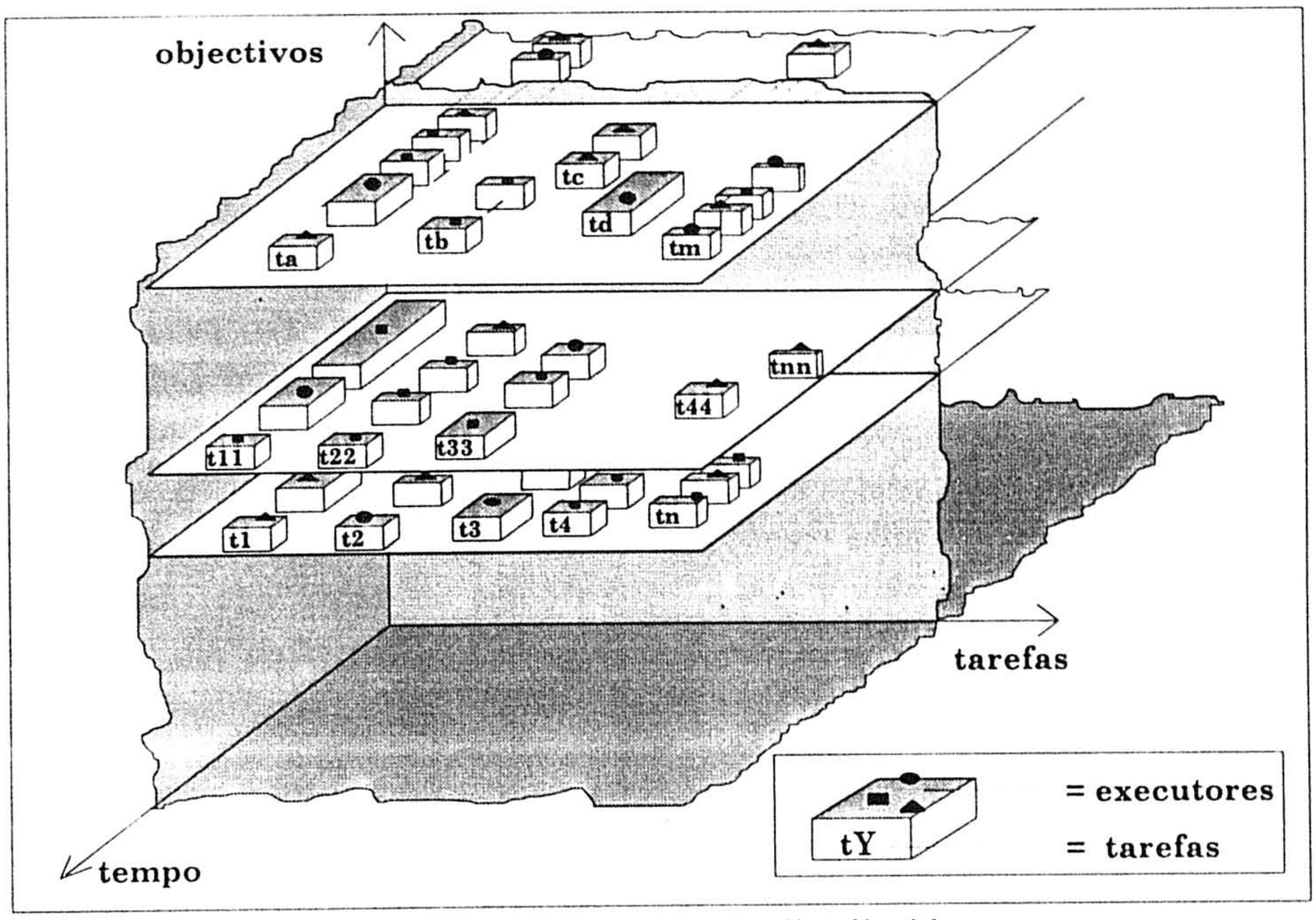
> (i) o que estuda a inteligência através das noções de pensamento e razão, com uma visão de cima para baixo, utilizando a introspecção e as experiências psicológicas como metodologias e,
> (ii) o que estuda a inteligência concentrando-se nos sistemas físicos, com uma visão de baixo para cima, através do conhecimento sobre o funcionamento dos sistemas biológicos.

A IA como uma área da Ciência da Computação, está interessada em projectar sistemas computacionais inteligentes [8]. Sistemas computacionais inteligentes são sistemas que apresentam características que nós associamos com a inteligência encontrada no comportamento humano (compreensão das linguagens de comunicação, aprendizagem, raciocínio, resolução de problemas, etc.), enquadrados numa ou mais das categorias citadas na [fig.4].

1.3.1 Inteligência Artificial Distribuída

Considerada como um dos ramos da Inteligência Artificial (IA) [9], a Inteligência Artificial Distribuída (IAD) está interessada em compreender e modelar acções e conhecimento de empreendimentos nos quais vários sistemas interagem para o alcance de um objectivo. A [fig. 5] ilustra graficamente um ambiente distribuído, onde cada objectivo é constituído por várias tarefas (**tY**), as quais ocorrem ao longo do tempo de forma sequencial e/ou paralela, e cujas execuções são da competência dos sistemas envolvidos (executores representados por •, ▯.. .). A compreensão e/ou a modelação do problema, objecto do estudo, caracteriza-se:

> (i)pela disposição distribuída da actividade e/ou da inteligência,
> (ii) pela decomponibilidade das tarefas e dos objectivos, e
> (iii) pela possibilidade de exploração do paralelismo na solução do problema.



[fig. 5] O ambiente distribuído

Duas questões básicas se colocam de forma a que os problemas do mundo real tenham uma abordagem através da IAD; uma é quanto à natureza dos sistemas envolvidos no problema em estudo e a outra é referente ao modelo adoptado para solução. As razões que levam à distribuição da actividade e da inteligência, em ambas as questões, são referentes:

(i) aos problemas serem fisicamente distribuídos. Actividade e inteligência estão espacialmente distribuídas, como por exemplo nas redes de transportes,

(ii) aos problemas serem funcionalmente distribuídos e heterogéneos. A complexidade de um problema a ser resolvido requer o seu fraccionamento. Este fraccionamento impõe uma visão local a cada componente, requerendo a participação de diferentes especialistas, os quais participam com a sua especialidade (domínio) na elaboração da solução. Como exemplo podemos citar a elaboração de projecto para a construção de edifícios,

(iii) ao incremento na rapidez da obtenção da solução do problema, através da exploração do paralelismo,

(iv) ao decremento da comunicação, enviando ou recebendo apenas as mensagens específicas e,

(v) a fiabilidade, isto é, à existência de alternativas quando um dos componentes não suporta a execução, ou falha.

Duas áreas fundamentais são distinguidas na IAD [9]:

(i) a área de Resolução Distribuída de Problemas ("*Distributed Problem Solving - DPS*") e,

(ii) a área de Sistemas Multiagente ("*Multiagent Systems - MAS*").

No caso da Resolução Distribuída de Problemas, esta está interessada em como a tarefa para solucionar um determinado problema pode ser dividida em

módulos (nós), os quais cooperam a nível de divisão e partilha do conhecimento sobre o problema e sobre o desenvolvimento da solução. Os nós, também denominados de componentes de "*software*", produzem resultados, os quais são recolhidos e compostos num resultado global [10], caracterizando uma coordenação intermódulos centralizada.

No caso da área dos Sistemas Multiagente, esta está interessada no comportamento individual e colectivo de unidades computacionais resolvedoras de problemas [11], as quais, através das suas capacidades, continuamente agem/reagem modificando o ambiente que as comporta. Caracterizadas pela ausência de uma coordenação centralizada das acções/reacções, estas unidades resolvedoras de problemas são chamadas de **agentes**, os quais no âmbito desta tese também são componentes de "*software*".

1.3.1.1 Sistemas Multiagente

1.3.1.1.1 Definições

De forma geral o termo sistemas multiagente tem sido aplicado a qualquer sistema composto de múltiplos agentes interactuantes [12]. De forma específica um sistema multiagente é um sistema composto de:

(i) um ambiente *Am*,
(ii) um conjunto *Ag* de agentes, e
(iii) um conjunto *Ob* de objectos, não agentes.

Um ambiente *Am*, é um espaço dinâmico, dotado de uma métrica que possibilita os agentes perceberem, localizarem e actuarem sobre os objectos. Um conjunto de agentes *Ag*, é um conjunto que apresenta uma estrutura de organização, de sociedade, de grupo ou micro, de agentes homogéneos ou heterogéneos, os quais agem/reagem no ambiente *Am* diante dos elementos do conjunto *Ob* ou face aos elementos do conjunto *Ag*, através de **comportamentos**. Um conjunto *Ob* de objectos é um conjunto de entidades, não agentes, presentes no ambiente *Am* que sofrem manipulações pelos elementos do conjunto *Ag*.

Um sistema multiagente tem uma estrutura de organização de **nível de sociedade** quando o foco se põe num grande número de agentes, seus múltiplos papeis, actividades e evolução na comunidade como um todo. É do **nível de grupo** quando o interesse está concentrado nas relações entre um número intermédio de agentes, seus papeis e actividades, agregados em torno de um segmento da sociedade. É do **nível micro** quando a ênfase se dá essencialmente nas relações entre dois ou entre um reduzido número de agentes, os quais representam um subconjunto de um grupo.

Uma variedade de definições de agentes [13] tem sido apresentada por investigadores da área da IAD, cada qual orientada aos seus trabalhos, com a finalidade de explicar o uso da palavra agente.

Em [6] encontramos:

"Um agente é uma entidade que pode perceber o seu ambiente através de sensores e agir sobre este ambiente através de actuadores".

Em [14] temos:

"Agentes autónomos são sistemas computacionais, os quais inseridos num ambiente dinâmico e complexo, percebem e actuam automaticamente neste ambiente, e fazendo-o, compreendem um conjunto de objectivos ou tarefas para as quais foram projectados".

Em [15] é apresentado:

"Definimos agente como uma entidade persistente de "*software*" dedicada a uma finalidade específica. O termo persistente é empregado para distinguir agentes de rotinas; agentes possuem suas próprias ideias de como devem realizar suas tarefas e as suas próprias agendas".

Em [16] temos:

"Agentes inteligentes executam de forma contínua três funções: (i) perceber as condições dinâmicas do ambiente, (ii) agir para afectar as condições do ambiente e, (iii) raciocinar para interpretar as percepções, resolver problemas, inferir e determinar acções".

Em [17] é apresentado:

"Agente é utilizado para denotar um sistema computacional que goza das seguintes propriedades: (i) autonomia: agentes operam sem a intervenção directa de seres humanos ou outras entidades, e exercem algum tipo de controlo sobre as suas acções e estados internos, (ii) habilidade social: agentes interagem com outros agentes, (iii) reactividade: agentes percebem os seus ambientes e respondem rapidamente às trocas que neles ocorrem e, (iv) pró-actividade: agentes não agem apenas em respostas às alterações dos seus ambientes. Eles são também capazes de exibir um comportamento orientado por objectivos através de iniciativas".

Em [27] temos a definição de uma espécie de agente chamado "*softbot - software robot*":

"Um "*softbot*" é um agente que interage com o ambiente computacional pela emissão de comandos e pela interpretação dos resultados apresentados pelo ambiente. Os actuadores dos "*softbots*" são comandos (comandos UNIX tais como "*mv*" ou "*compress*") os quais possuem como objectivo promover alterações no estado do ambiente. Os sensores dos "*softbots*" são comandos (comandos UNIX tais como "*pwd*" ou "*ls*") os quais possuem como objectivo produzir informação".

Em [108] encontramos os agentes móveis "*mobile agents*":

O conceito de agentes móveis emergiu, na década de setenta, de um exame profundo da comunicação entre computadores. Devido ao crescimento exponencial da "*INTERNET*", uma nova aproximação foi necessária para incrementar os tipos sofisticados de comunicação (protocolos), sem congestionar os componentes da mesma.

O conceito actual, em termos de comunicação para redes de computadores, é o "Chamado de Procedimento Remoto ("***Remote Procedure Calling***")". O "Chamado de Procedimento Remoto" é baseado na solicitação, de um computador à outro, para a execução de procedimentos. As mensagens entre computadores são pedidos ou confirmações das execuções de procedimentos. Um pedido inclui dados que são os argumentos do procedimento e a resposta inclui dados que são resultados da execução do procedimento.

Os computadores envolvidos, em um "Chamado de Procedimento Remoto", desempenham os papéis de cliente e de servidor, dependendo de quem solicita e de quem atende a solicitação, respectivamente. Como exemplo, em uma solicitação para exclusão de todos os ficheiros com mais de um mês sem utilização, o computador-cliente deverá emitir duas chamadas:

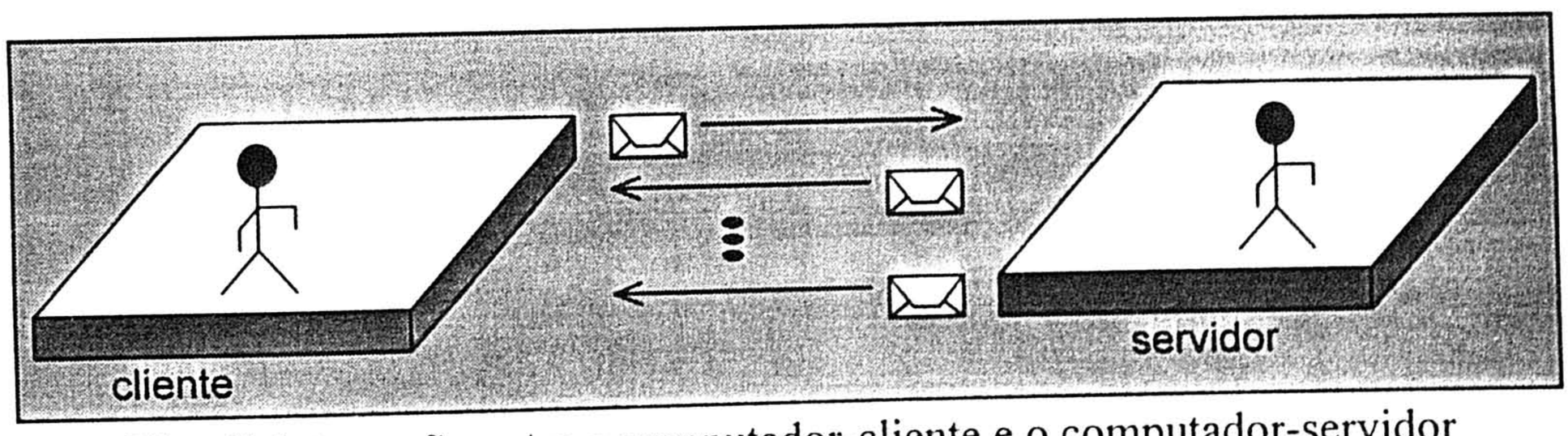
(i) uma para obter a identificação e o tempo de utilização dos ficheiros, e

(ii) a segunda, após aplicação do critério (executado no computador-cliente), a ordem do(s) ficheiro(s) a ser(em) excluído(s). Se houver a decisão de exclusão de "***n***" ficheiros, o número de mensagens que o computador-cliente deverá emitir ou receber será de $2(n+1)$.

O "*RPC*", essencialmente, requer que para cada interacção entre o computador-cliente e o computador-servidor, existam dois actos de comunicação [fig. 6]:

(i) um para perguntar ao servidor para executar um procedimento, e

(ii) outra para confirmar que o servidor executou o procedimento, o que conduz a que cada série de interacções entre computadores, requeira uma série similar de comunicação.



[fig. 6] Interacção entre o computador-cliente e o computador-servidor

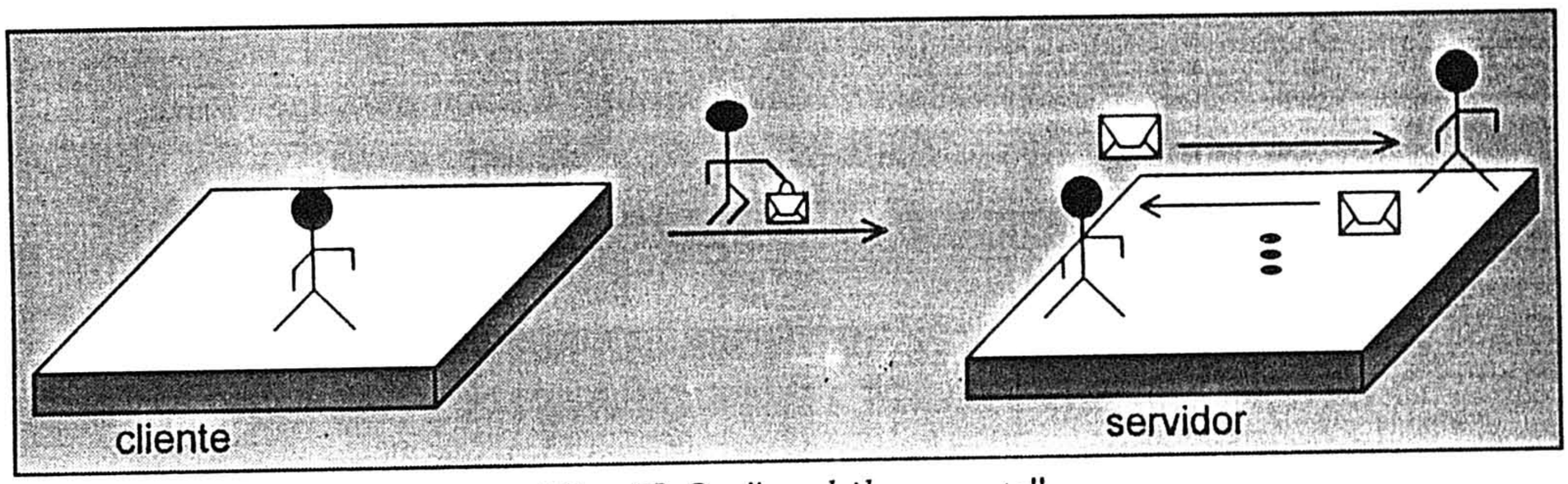
Uma alternativa ao "*RPC*" é a "Programação Remota ("***Remote Programming***")" cuja comunicação computador a computador não é apenas a de um computador chamar procedimentos em outro, mas o de supri-lo de procedimentos a serem executados. Os procedimentos e seus estados são denominados de agentes móveis ("*mobile agents*") [fig. 7]. Estes agentes representam o computador que os enviam, e

operam no computador que os recebem. No mesmo exemplo, do pedido de exclusão de ficheiros, não importa o número de ficheiros a ser excluído, é requerido apenas a mensagem que transporta o agente entre computadores. Utilizando a "*RP*", o computador-cliente e o servidor podem interagir sem utilização da rede, uma vez que a rede já transportou um agente entre eles.

A "*RP*" apresenta duas vantagens fundamentais em relação a "*RPC*":

(i) a quantitativa e táctica (desempenho): quando um computador-cliente tem uma tarefa para um servidor, em vez de enviar comandos, através da rede, ele envia um agente que passa a executar o trabalho localmente e não remotamente, e

(ii) a qualitativa e estratégica (personalização): a vantagem estratégica apresentada pela "*RP*" é a de possibilitar aos desenvolvedores de "*softwares*" para utilizadores, estenderem as funcionalidades dos servidores.



[fig. 7] Os "*mobile agents*"

Os três sistemas mencionados a seguir, são exemplos de sistemas de agentes móveis:

(i) "*ARA*" - ("*Agents for Remote Access*"). O "*ARA*" é um sistema para desenvolver plataformas de "*softwares*" baseadas em agentes. Estes agentes são capazes de se movimentarem, livremente, quando eles assim decidirem, sem interferirem nas execuções de outros procedimentos. Os movimentos podem ocorrer entre computadores dotados de diferentes recursos.

(ii) "*TCL*" - ("*Tool Command Language*"). O "*TCL*" é baseado em uma plataforma de mesmo nome, desenvolvida pela "*Sun Microsystems*", o qual é um complemento para a sua linguagem de programação "*Java*".
Os agentes "*TCL*" possuem quatro objectivos:

(i) reduzir a movimentação para uma simples instrução "*agent_jump*", e permitir que esta instrução ocorra em qualquer ponto da rede de computadores,
(ii) providenciar a flexibilidade e o baixo nível de comunicação,
(iii) providenciar uma linguagem de alto nível, e
(iv) providenciar uma efectiva segurança.

(iii) "*Aglets Workbench*"
É um sistema de agentes móveis desenvolvido pela "*IBM*" do Japão, baseado em programação "*Java*". O "*Aglets Workbench*" apresenta ao utilizador um ambiente visual para a elaboração de soluções que utilizam agentes móveis na pesquisa, no acesso e na gestão de dados corporativos e outras informações. As duas principais características do "*Aglets Workbench*" são:

(i) o ambiente visual que permite ao utilizador, rapidamente e facilmente, criar agentes para a "*INTERNET*" e seus componentes, capacitando-os a terem acesso a bases de dados corporativas, e
(ii) a pesquisa, movimentação e comunicação de maneira padrão e segura.

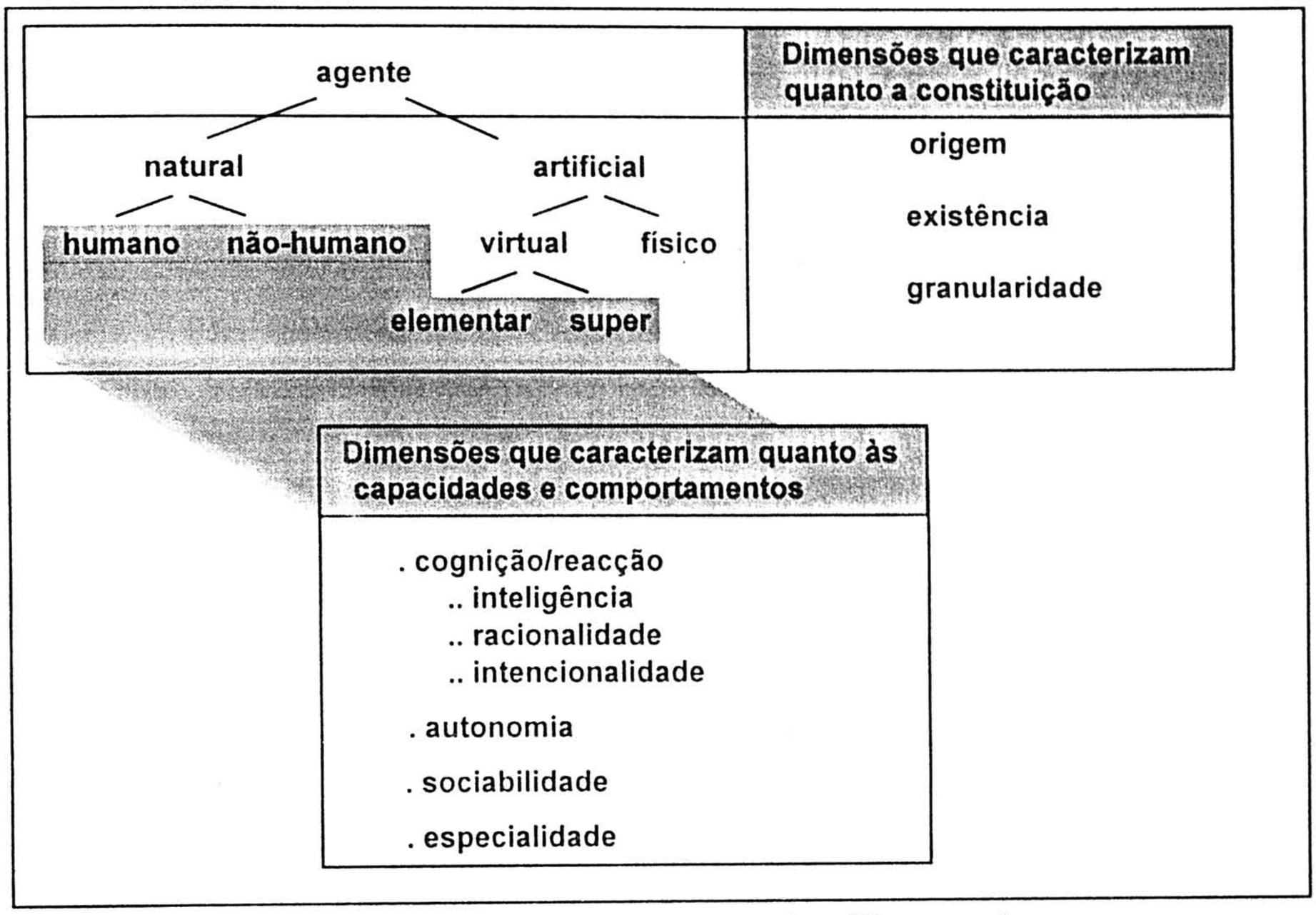
O "*Aglets Workbench*" providencia a possibilidade de criar programas baseados em "*Java*", os quais podem ser disparados de um computador e transportados para um outro remoto, para a execução. No computador remoto, os agentes móveis apresentam suas credenciais e obtém acesso a dados e a serviços locais. O computador remoto pode também trazer para junto de si, agentes com interesses similares para compartilharem os objectivos, providenciando um local de encontro no qual os agentes possam interagir.

1.3.1.1.2 Classificações

Não só as definições acima citadas, bem como as definições em geral, apresentam propriedades as quais, além de permitirem a distinção e caracterização de entidades, no caso agentes, em relação a outras entidades, possibilitam a formação de classes. Estas classes são baseadas em dimensões (concepção, constituição, granularidade, etc.), e são úteis para o enquadramento dos agentes, permitindo classificar o sistema multiagente em sistema **heterogéneo** ou **homogéneo** dependendo da dimensão em análise.

Um sistema multiagente é homogéneo quando os agentes participantes possuem, nas dimensões em análise, valores aceitáveis e congruentes. Um sistema multiagente é heterogéneo quando os agentes participantes possuem, nas dimensões em análise, valores aceitáveis e não congruentes. A [fig. 8] apresenta um de entre vários esquemas possíveis [109] de classificação. Este esquema não é exaustivo, na

medida em que apresenta apenas algumas das dimensões possíveis para uma completa taxonomia de agentes. Contudo auxilia em geral na compreensão do que são agentes, e em particular na compreensão dos agentes tratados por esta dissertação.



[fig. 8] Algumas das dimensões para classificação de agentes

Em referência à [fig. 8], temos as dimensões que caracterizam os agentes quanto à constituição e quanto as capacidades e comportamentos.

Quanto à constituição dos agentes:

(i) origem: enquadra os agentes quanto à concepção, podendo ser: natural ou artificial. É natural quando o agente está contido nos objectos ou fenómenos estudados pelas ciências naturais. É artificial quando é um artefacto "*man-made*" [1],

(ii) existência: os agentes artificiais, quanto à sua existência no mundo real, podem ser: física ou virtual. É física quando altera o mundo real (ex. robô, automóvel, etc.). É virtual quando se trata de um componente de "*software*" que representa um agente, e

(iii) granularidade: enquadra os agentes em termos de constituição quantitativa, podendo ser: elementar ou super. É elementar quando os componentes do agente não são qualificados como agentes. É super quando o agente é constituído por parte menores as quais são agentes elementares ou super agentes.

Quanto as capacidades e comportamentos dos agentes:

(i) cognição/reacção: enquadra os agentes em termos de capacidades de resolução de problemas. São cognitivos se dotados de observação, raciocínio e memória e capazes de criar e executar planos de acção para o alcance de objectivos. Nesta perspectiva esses agentes são ditos inteligentes podendo ser qualificados de racionais. Entende-se

como racionais se agem em busca de seus objectivos da forma certa (dada uma medida de sucesso na tarefa a ser executada, buscam a mais alta graduação), e não agem de forma a evitá-los. Os cognitivos são chamados de intencionais na medida em que a estrutura de conhecimento para a aplicação do raciocínio se caracteriza pela existência de crenças, desejos e intenções ("*BDI-architecture, Beliefs, Desires and Intentions*"). Esta estrutura do conhecimento expressa uma atitude mental do agente [18], representando respectivamente a informação, a motivação e o raciocínio (referidos também na literatura como agentes deliberativos).

São reactivos na medida em que percebem o ambiente no qual estão inseridos e respondem rapidamente [28] às trocas que nele ocorrem. O termo rapidamente está associado à ideia de uma baixa actividade de raciocínio (permitindo reacções rápidas) o que leva a caracterizá-los como incapazes de manipularem os seus objectivos, i.e. suas acções são executadas como o resultado de disparos de regras simples, dado pelo par (percepção, reacção). Quanto à inteligência dos agentes reactivos, esta emerge não do comportamento individual, mas do comportamento global da comunidade,

(ii) autonomia: enquadra os agentes em termos do exercício do controlo sobre as suas acções [20]. Pode variar de forma contínua desde completamente autónomo, quando o controlo está a cargo do agente referido até totalmente não autónomo, quando o controlo se encontra fora deste,

(iii) sociabilidade: enquadra os agentes em termos de possuírem explicitamente ou não, modelos dos outros agentes da comunidade, e através do raciocínio, considerarem este conhecimento para a tomada de decisão, e

(iv) especialidade: enquadra os agentes em termos do domínio das tarefas que os mesmos executam.

Em [19] distingue-se ainda o uso do termo agente denotado por características fracas e por características fortes. São características fracas para um agente as seguintes propriedades: autonomia, sociabilidade, reactividade e pró-actividade (cognição). São consideradas características fortes quando, além de possuírem as fracas, possuem uma conceptualização ou uma implementação, as quais empregam conceitos usualmente aplicáveis a seres humanos. Estes conceitos dizem respeito a notações de estados mentais tais como; conhecimento, crenças, intenções (sistemas "***Beliefs, Desires and Intentions***" Anexo A), obrigações e emoções .

1.3.1.1.3 Implementações

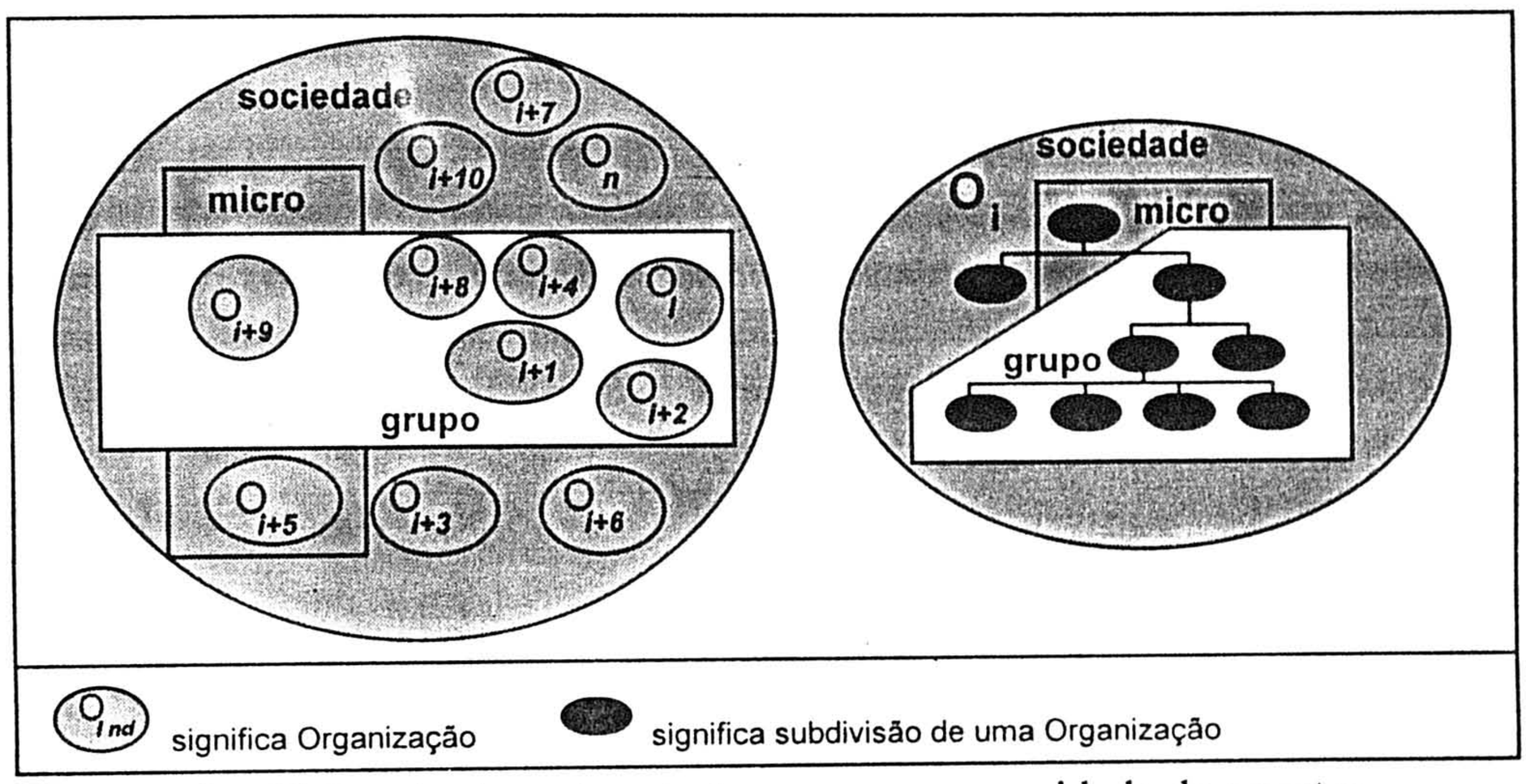
Quanto à implementação de sistemas multiagente há que considerar as perspectivas relativas à comunidade de agentes como um todo, a qual actua na resolução do problema do mundo real que se está a modelar (cenário) e aos agentes

propriamente ditos. As perspectivas relativas à comunidade de agentes dizem respeito à:

(i) estrutura da organização desses agentes,
(ii) interface homem/sistema multiagente, e
(iii) comunicação interagentes (a interface agente/agente).

As perspectivas relativas aos agentes propriamente ditos dizem respeito às propriedades, arquitectura e linguagem de programação que cada um destes apresenta.

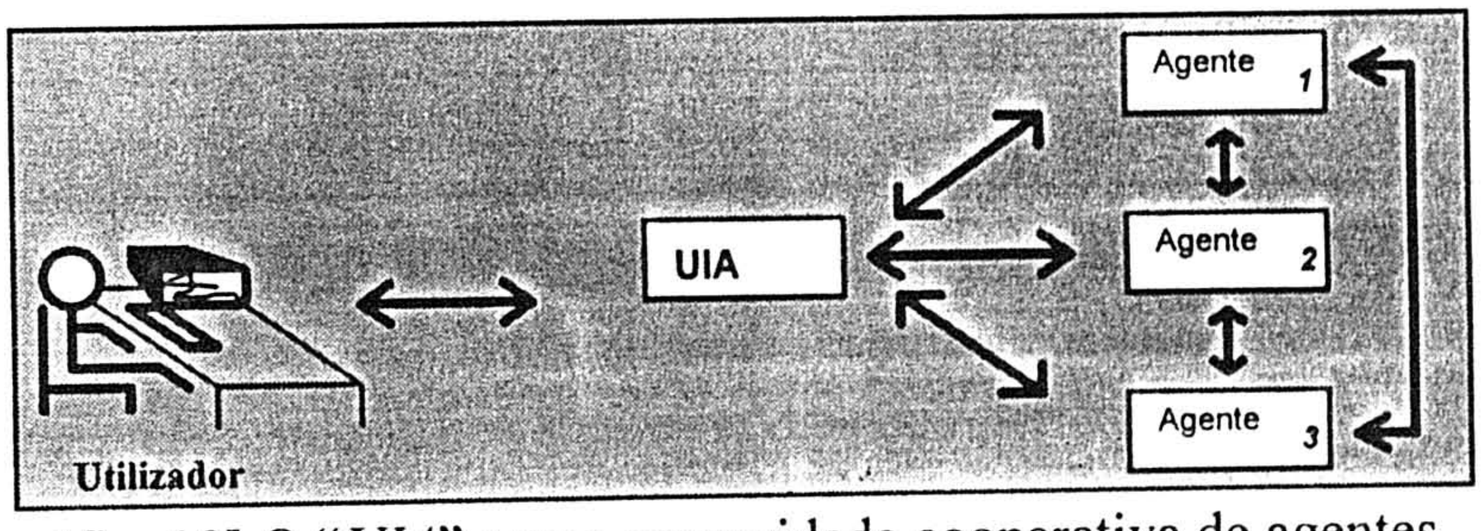
Na questão da estrutura da organização da comunidade de agentes, esta apresenta-se em função da abrangência da relação que é estabelecida interagentes. A [fig. 9] apresenta os três tipos de estruturas, tendo como exemplo uma sociedade de organizações (organizações produtoras de bens/serviços, com ou sem fins lucrativos).



[fig. 9] Os três tipos de estruturas numa comunidade de agentes

Utilizando como referência a relação de contratação de colaboradores para o fornecimento de um bem ou serviço, a qual iremos discorrer nos próximos capítulos, o nível de sociedade é a totalidade acessível e observável das organizações de um universo, quando um convite de participação para um processo de selecção é feito. São do nível de grupo aquelas organizações que aceitam participar (respondem ao convite). Cabe ressaltar que os critérios para segmentação em grupos podem ser outros que não apenas o aqui referido. O nível micro surge nas relações específicas entre duas ou entre um número reduzido de organizações, que representam por exemplo as organizações que ficaram para uma fase final do processo selectivo. Ainda com referência à [fig. 9] notam-se a existência de duas sociedades. Isto ocorre devido a, na nossa abordagem de sistemas multiagente, estarmos a modelar os agentes que representam as organizações de uma sociedade de organizações, como agentes do tipo artificial, virtual e super/elementar, enquadrados segundo os critérios de origem, existência e granularidade respectivamente. A importância das estruturas de organização de um sistema multiagente faz-se notar nas relações interagentes. Como se trata de relações sociais (relações que consideram o conhecimento dos demais agentes envolvidos), seja para efeito de raciocínio, de comunicação ou aprendizagem, a abrangência da comunidade em causa, torna-se factor de sucesso ou não das actividades cooperativas.

Uma interface homem/sistema multiagente é um sistema capaz de apresentar ao ser humano o comportamento da comunidade de agentes ou de agentes específicos, através de uma linguagem simbólica compreensível ao mesmo, com o objectivo de informar ou permitir a interacção com a comunidade de agentes. O ser humano passa a ser um agente (agente natural humano) passivo ou activo do sistema multiagente. É passivo quando o ser humano é observador e colector de informações a serem utilizadas em outro sistema. É activo quando interage com os componentes do sistema multiagente, o que leva os demais elementos a considerarem a sua actuação. Em [21] encontramos a definição de agentes actuando como uma interface: são programas de computador que empregam técnicas de inteligência artificial com o objectivo de auxiliar o utilizador numa aplicação particular. Trabalhos relativos a este tópico são também encontrados no enquadramento do Trabalho Cooperativo Auxiliado por Computador ("*CSCW - Computer Supported Cooperative Work*"). A área de "*CSCW*" [22] está fundamentalmente interessada no desenvolvimento de ferramentas de "*software*" que possam auxiliar no trabalho cooperativo humano. Proposto por [23], e designado de Sistemas Participantes ("*Participant Systems*"), esta proposição apresenta a colaboração não apenas entre seres humanos (agentes naturais humanos) mas também empregando agentes artificiais. Em [10] é apresentado um agente de interface com o utilizador ("*User Interface Agent - UIA*") [fig. 10], o qual integra o utilizador (neste caso um operador num centro de controlo de uma rede de distribuição de energia eléctrica) num sistema multiagente (gestor automático da rede).



[fig. 10] O "*UIA*" numa comunidade cooperativa de agentes

1.3.1.1.4 Interacções

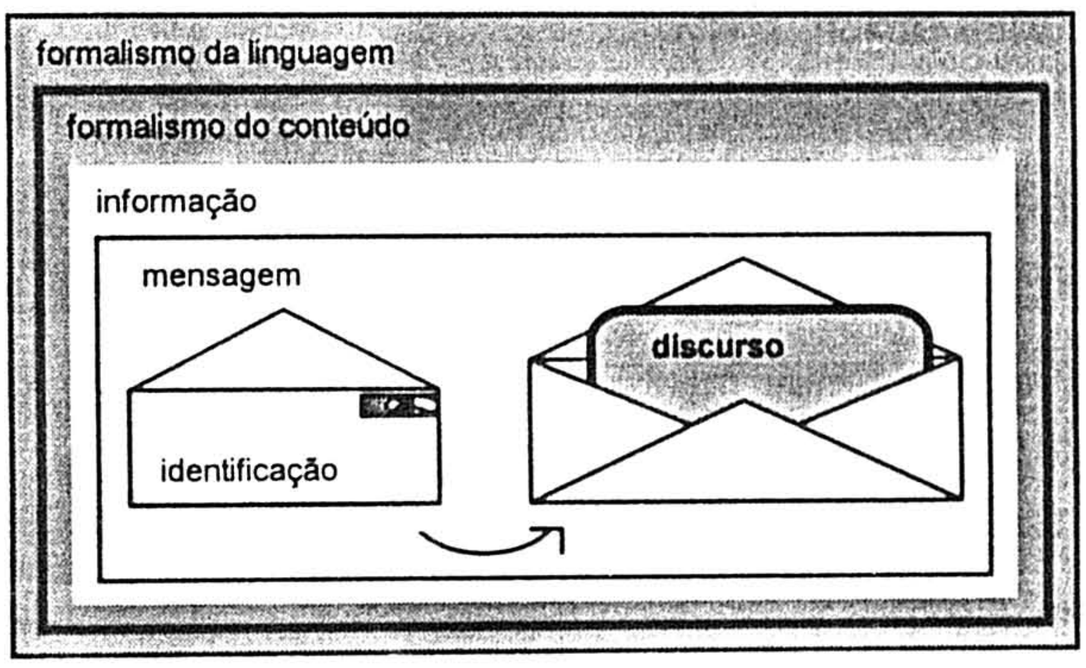
As interacções de agentes num ambiente *Am* são realizadas através de acções/reacções baseadas numa linguagem, o que é designado de comunicação, ou através de acções/reacções não baseadas em linguagens (caminhar, destruir, mover, etc.). As linguagens possibilitam o envio e o recebimento de informações entre os agentes de um sistema multiagente [24], permitindo a coordenação das suas actividades. Como suporte à comunicação em sistemas multiagente, duas estratégias podem ser utilizadas: ou as mensagens são trocadas directamente (1:1,1:n ou n:n) entre agentes, ou são disponibilizadas num **quadro-negro** ("*blackboard*") permitindo o acesso a elas. Em [25] encontramos a definição de quadro-negro como uma base de dados global em torno da qual todas as actividades para a resolução de um problema

são concentradas. No domínio desta dissertação estaremos a utilizar as técnicas de quadro-negro como repositório de informações à comunidade de agentes. O repositório representa um local de afixação (quiosque) de notícias tendo as informações como representantes de anúncios aos agentes que estejam interessados em participar de um processo de selecção. Os agentes interessados são os que possuem como um dos seus objectivos a constante procura de oportunidades de negócios para fornecimento de um bem/serviço aos demais agentes representativos das organizações.

As interacções entre agentes baseadas em comunicação [fig. 11], apresentam dois formalismos. Um é relativo à linguagem (linguagem formal) na qual, a informação a ser trocada com a comunidade de agentes está representada (por exemplo a lógica de primeira ordem). O outro é relativo à expressão do conteúdo da informação. A expressão do conteúdo da informação é dividida em duas partes:

(i) a identificação da origem/destino da mensagem, a qual contém informação sobre a localização do agente emissor e do receptor, e

(ii) o conteúdo da mensagem, que apresenta a mensagem propriamente dita (tipo de acto de discurso e o discurso).



[fig. 11] A composição de uma mensagem

O exemplo a seguir apresenta o esquema das expressões da linguagem de comunicação (designadas por mensagens) intra e interagentes citadas em [10].

Exemplo:

mensagem(<tipo_de_mensagem>, <id_nós>, <informação>)

Esta mensagem possui um formalismo de linguagem em "*Prolog*", cujo formalismo de conteúdo em BNF ("*Bachus-Naur Form*") é:

<expressão>::=
mensagem,
"(",<tipo_de_mensagem>,",",<id_nós>,",",<informação>,")"

O não-terminal <*tipo_de_mensagem*> é formado por uma ou duas palavras chaves:

```
<tipo_de_mensagem>::=<tipo>,",",<subtipo>
<tipo_de_mensagem>::=<tipo>
```

A expressão <*id_nós*> é constituida pelos nó emissor e nó receptor, só pelo nó emissor ou então não existe. Nestes dois últimos casos a identificação do nó ou nós não presentes, deve poder ser deduzida inequivocamente.

```
<id_nós>::=<nó_emisor>,",",<nó_receptor>
<id_nós>::=<nó_emissor>
<id_nós>::=
```

O não_terminal <*informação*> tem, num caso geral, uma parte baseada na linguagem do conteúdo, e outra parte bastante dependente do tipo específico de nós para os quais a linguagem foi desenvolvida.

Na instância que se segue:

```
mensagem( volunteer, Ag_origem, Ag_destino, Conteúdo, Valor_lógico)
```

(i) *volunteer*, *Conteúdo* e *Valor_lógico*, referem-se a parte do discurso, e
(ii) *Ag_origem* e *Ag_destino* referem-se a parte de identificação.

O anexo B apresenta, em mais detalhes, a linguagem de comunicação intra e interagentes citada em [10], e a linguagem "*KQML*".

Para a implementação de agentes "artificiais-virtuais" constituir um sistema multiagente, questões como:

(i) quais são as propriedades desejadas?
(ii) qual a arquitectura? e
(iii) qual a linguagem de programação?

devem ser consideradas.

Quanto às propriedades, e com base no cenário (problema do mundo real a ser modelado), deve dotar-se os agentes de:

(i) dimensões (classes) e de valores para estas dimensões,
(ii) uma estrutura de conhecimento e sua forma de manutenção, e
(iii) capacidades de percepção, geração de acções e/ou reacções, adaptação e aptidões de aprendizagem.

O anexo C apresenta as propriedades gerais de agentes e as propriedades específicas para agentes deliberativos.

1.3.1.1.5 Arquitecturas

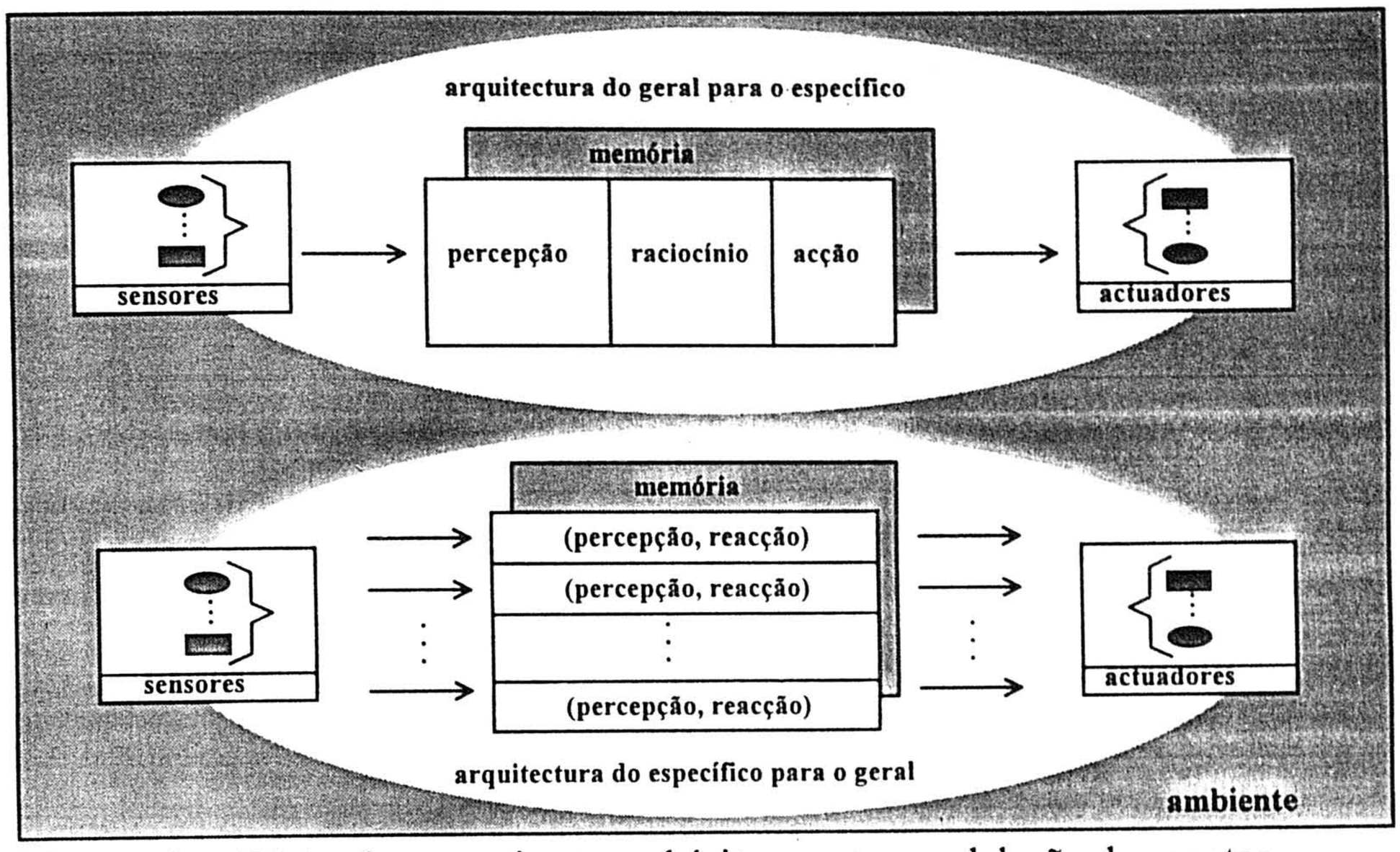
Quanto à arquitectura dos agentes, ela está orientada para satisfazer as propriedades requeridas.

Uma arquitectura sugere uma metodologia [26] para construir agentes , a qual:
(i) especifica como o problema pode ser decomposto em subproblemas,
(ii) como cada subproblema dá origem a módulos (técnicas e algoritmos), e
(iii) como se deve implementar estes módulos para interagirem.

O conjunto de módulos e suas interacções providenciam uma resposta para a questão de como os objectos e fenómenos observados no ambiente e o corrente estado interno do agente, determinam as acções e os futuros estados internos do mesmo.

Duas aproximações para a modelação de agentes tem sido empregadas:
(i) a abordagem do geral para o específico ("*top-down*"), e
(ii) a abordagem do específico para o geral ("*bottom-up*").

Agentes projectados segundo a abordagem do geral para o específico, caracterizam-se pela modularização das habilidades cognitivas. Isto significa que o modelo geral é inicialmente desenvolvido, seguido das implementações separadamente dos diferentes componentes. Por outro lado agentes projectados segundo a abordagem do específico para o geral, contemplam inicialmente a implementação de comportamentos simples (cobrindo uma completa unidade de percepção e reacção) e então incrementalmente, comportamentos mais sofisticados são adicionados. A [fig. 12] apresenta as arquitecturas básicas para estas duas aproximações.



[fig. 12] As duas arquitecturas básicas para a modelação de agentes

Na arquitectura referida como "do geral para o específico", o agente raciocina utilizando a lógica baseada em manipulação simbólica e a adaptação de padrões ("*pattern matching*"). Toma decisões com base num modelo simbólico explícito do mundo (ambiente e estados mentais) contido na memória, gera planos de acção e executa-os.

A arquitectura referida como "do específico para o geral", é baseada na ideia de que muitas das nossas actividades diárias consistem em acções rotineiras. Neste contexto o agente é projectado de forma a conter um conjunto de comportamentos simples os quais reagem às mudanças detectadas no ambiente, na modalidade "(percepção, reacção)". Isto resulta numa arquitectura onde os aspectos cognitivos são reduzidos a um mapeamento dado pelo par "(percepção, reacção)".

Em ambas as arquitecturas as percepções e as acções/reacções são, respectivamente, baseadas em sensores e actuadores.

Quando a arquitectura combina as duas aproximações citadas, é nomeada de arquitectura híbrida. Desta forma, dota-se o agente com a habilidade para reagir, a qual suporta as tarefas rotineiras sem recorrer a raciocínios complexos, e com capacidades de raciocinar e agir para tarefas de longo prazo ou tarefas que requeiram maior quantidade de considerações (deliberações).

O anexo D apresenta alguma das arquitecturas para agentes deliberativos, reactivos e híbridos.

1.3.1.1.6 Linguagens

Em relação à linguagem de programação, esta participa como uma das perspectivas do mundo dos agentes, na medida em que sustenta a implementação dos agentes a partir das propriedades e da arquitectura.

Algumas dessas linguagens já incorporam uma forte noção dos conceitos de agente, possibilitando primitivas para a construção dos mesmos, tais como (Anexo E):

(i) programação orientada a agentes ("*agent-oriented programming*") [110],

(ii) as linguagens "*April*" e "*Mail*" desenvolvidas no âmbito do projecto "*ESPRIT-IMAGINE*" [111], e

(iii) a já comercial "*TELESCRIPT*" ("*General Magic, Inc*" [112]).

Outras, de propósitos mais gerais, suportam o desenvolvimento não só do mundo dos agentes, como também de aplicações genéricas, tais como:

(i) linguagem orientada à objectos ("*object-oriented language - OOL*"), tendo como exemplos: "*C++*", "*Java*" [113], "*Smalltalk*", etc,

e

(ii) programação em lógica ("*PROLOG*" [92]),

entre outras.

Plataformas computacionais, tal como "*CORBA*" [116], para facilitar a comunicação entre processos (os quais poderão ser agentes) estão disponíveis.

1.3.2 Aprendizagem

A aprendizagem (ou acto de aprender) está fortemente vinculada ao significado da palavra mudança. A aprendizagem pode ser vista como o acto de promover mudanças úteis nos estados mentais, as quais levam um agente (aprendiz) a desempenhar alguma actividade (mental/física) até então por este desconhecida ou a desempenhar de forma mais eficiente/eficaz uma já presente no seu domínio de conhecimento.

Quanto à utilidade das mudanças, estas podem temporariamente não apresentar as características de benefício/importância individual ou social (uma aparente não utilidade), todavia elas podem contribuir para a aprendizagem ou execução de outras actividades, que não a do domínio inicial, provocador das mudanças. Por exemplo, qual seria a utilidade da mudança provocada pelo processo de aprendizagem no domínio do acondicionamento de objectos numa loja de brinquedos, se o potencial aprendiz não trabalha nem tem a intenção de trabalhar na actividade de fazer embrulhos? Aparentemente a resposta é: nenhum interesse! Porém, pelo menos duas razões podem ser apresentadas, contrariando a resposta:

(i) o ser humano é um ser dinâmico, bem como o ambiente no qual vive, o que pode levá-lo ao longo do tempo a passar a ter interesse ou necessidade de tal, e

(ii) uma vez o aprendiz tendo que desempenhar o papel de pai e com a necessidade de auxiliar na troca de fraldas do bebé, a adaptação do processo de embrulhar brinquedos pode ser útil para a troca das fraldas.

Com este exemplo não se quer afirmar que o que pode aparentemente não ser útil hoje, necessariamente será útil amanhã. Se quer apenas colocar a questão que a

utilidade não pode ser vista apenas num momento no tempo, mas ao longo deste, e que, a aprendizagem estão relacionados outros termos, tais como: generalização, prática ou aplicação, conhecimento, memória e associação.

1.3.2.1 Um Pouco de História sobre a Obtenção e Origem do Conhecimento

Uma das formas mais simples de aprendizagem é a **habituação**. Ela consiste:
(i) na diminuição da tendência para responder aos estímulos que se tornaram familiares, e
(ii) no efeito das respostas a estes estímulos se tornarem "automáticas",
devido a uma exposição repetida aos mesmos. Trata-se de uma reacção adaptativa.

Com a habituação, o organismo aprende a reconhecer um acontecimento como algo familiar, mas nada aprende sobre a relação entre este acontecimento e quaisquer outras circunstâncias.

No condicionamento clássico as relações passam a ser alvo de atenção e são chamadas de **associações**, cujo trabalho experimental se iniciou com Ivan P. Pavlov (1849-1936), o qual é conhecido por reflexo condicionado. O procedimento consistia de, no lugar de esperar pela ocorrência acidental dos acontecimentos na história do animal, o experimentador é que produzia tais acontecimentos.

Seguida destas duas formas de aprendizagem encontramos a **aprendizagem instrumental** (também chamada de condicionamento operante). Esta forma, mesmo iniciada antes das experiências de Pavlov, teve expressividade com o trabalho de Edward L. Throndike (1874-1949). Ela consistia em levar o animal a executar uma determinada acção estabelecida pelo treinador, em troca de uma recompensa.

Dentro desta mesma linha de pesquisa, encontra-se B. F. Skinner, o qual insistiu na ideia de uma distinção nítida entre o condicionamento oriundo dos estudos de Pavlov e de Throndike. O seu enunciado preconiza que, em oposição ao condicionamento clássico onde o comportamento do animal é evocado pelo estímulo vindo do exterior, as reacções são emitidas do interior, como se fossem voluntárias, designando estas respostas instrumentais de operantes; operam no ambiente para realizar uma mudança que leva à recompensa.

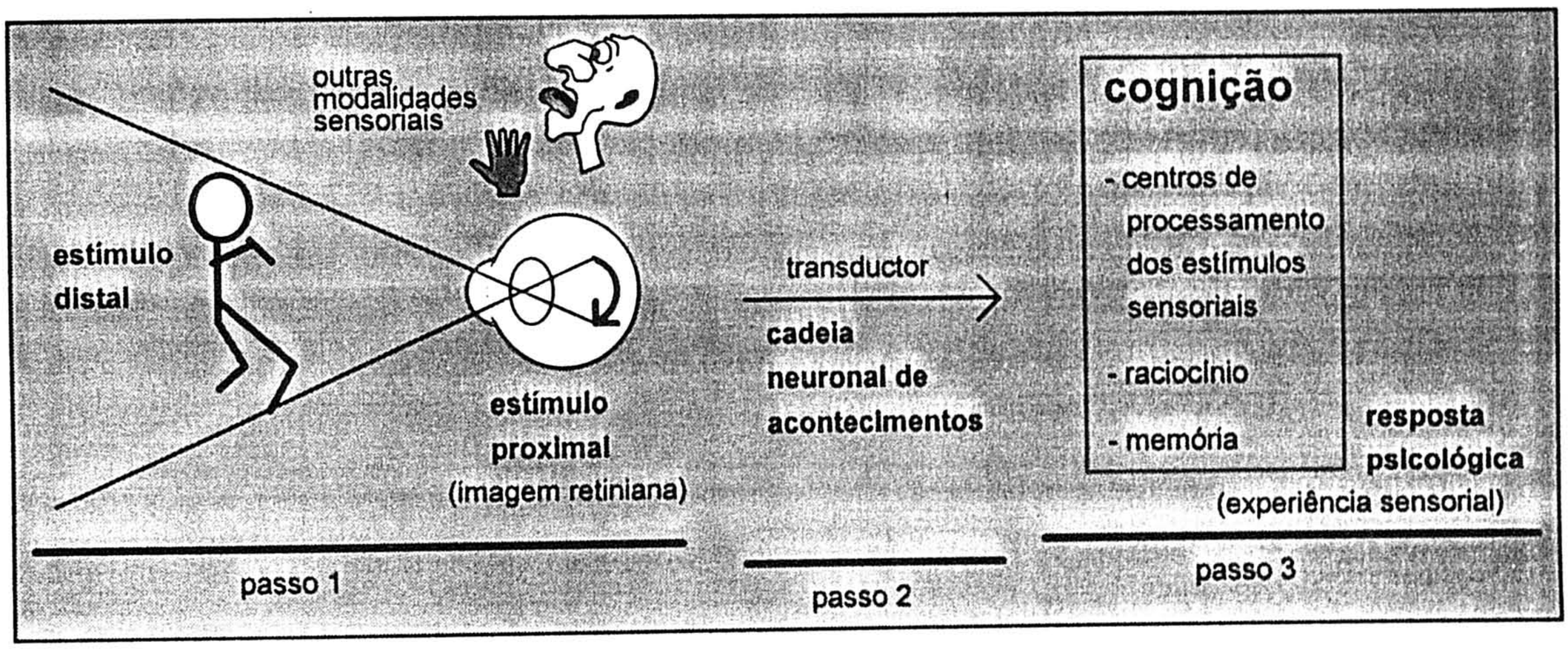
Ao lado das teorias baseadas em comportamento como as acima mencionadas, encontramos a aprendizagem chamada de **cognitiva**. Para os teóricos cognitivistas, a essência do que se aprende está dentro do animal e, sendo um acontecimento reservado, só se conhecerá que o animal aprendeu quando este actuar/reagir em conformidade com o novo estado mental. Este novo estado mental do aprendiz diz respeito a alterações no conhecimento com o surgimento de relações conceituais abstractas.

Vinculado à aprendizagem, encontra-se o **conhecimento**, cuja origem sempre foi tema de discussão e de investigação. Duas escolas, a **empirista** e a **inatista**, mesmo de posições opostas entre si, oferecem uma complementaridade para a compreensão deste fenómeno. Os empiristas concentram-se na ideia de que o conhecimento é adquirido a partir de experiências. John Locke (1632-1704), um dos

principais proponentes desta escola, defendia que todo o conhecimento é proveniente dos sentidos, não havendo ideias inatas. A mente seria como uma tábua rasa na qual a experiência iria imprimir as sua marcas.

Por outro lado, os inatistas, observam que vários aspectos do nosso conhecimento têm origem em características do cérebro (cujas raízes remontam a Platão). Uma das respostas modernas mais influentes ao empirismo foi-nos apresentada por Immanuel Kant (1724-1804), o qual defendia que o conhecimento não pode provir apenas dos dados sensoriais, tendo que existir certas categorias preexistentes (sistema nervoso), em torno das quais esses dados sensoriais são organizados. As experiências funcionam como fornecedoras de dados sensoriais, os quais são estruturados segundo o sistema nervoso inato.

Independentemente da intensidade com que cada uma destas escolas contribuíram para a formulação da obtenção do conhecimento, a integração da informação genética (inatista) e da ambiental (empirista) resulta numa perspectiva psicofísica entre as características dos estímulos físicos e a experiência psicológica a que estes estímulos dão origem. A [fig. 13] apresenta três passos básicos (modelo simplificado) para que uma sequência de acontecimentos se inicie num estímulo e termine num conceito do objecto observado.



[fig. 13] Sequência de acontecimentos a partir do estímulo até a formação de um conceito

A organização dos detalhes nesta sequência é diferente considerando as várias modalidades sensoriais (tacto, olfacto, audição, paladar e visão). Porém, de forma geral, no primeiro passo da sequência encontra-se o estímulo proximal (no exemplo, a imagem que é formada na retina) a partir do estímulo distal (objectos e eventos reais do ambiente observado). Como segundo passo é desencadeada uma cadeia neuronal de acontecimentos à qual o estímulo dá origem (o estímulo é convertido para uma linguagem compreendida pelos neurónios). Por fim, ocorre uma resposta psicológica à mensagem, a qual apresenta, frequentemente, a forma de uma experiência sensorial consciente (uma sensação).

1.3.2.2 Aprendizagem Automática

A disciplina de Aprendizagem Automática ("*Machine Learning*"), é uma das áreas da Inteligência Artificial, que tem como um dos seus principais objectivos, o estudo e a compreensão dos processos de aprendizagem bem como o desenvolvimento de algoritmos que suportem estes processos, para aplicação em artefactos. De forma geral estes algoritmos devem proporcionar mudanças de estados mentais, que venham a melhorar o desempenho dos artefactos (em particular os agentes), na execução de actividades, ao longo do tempo. O termo melhorar, significa apresentar um comportamento mais eficiente/eficaz (menor tempo de execução da actividade, acréscimo de confiança das relações entre agentes, obtenção de perícia para novas actividades ou relações no domínio de actuação do artefacto ou fora deste, alcance satisfatório de um objectivo, etc.) no desempenho de actividades.

Na história da aprendizagem automática três paradigmas da sua evolução destacam-se:

(i) o modelamento **neuronal** e as técnicas de **decisão**.

No modelamento neuronal o interesse recai na construção de sistemas de aprendizagem de propósito geral, cujo processo de aprendizagem consiste em trocas incrementais das probabilidades nos elementos que representam os neurónios numa rede neuronal artificial.

Como exemplos de trabalhos pioneiros na aprendizagem computacional destacam-se o "*Perceptron*" [29] e o "*Pandemonium*" [30, 31]. Nas técnicas de decisão, dentre outros, o trabalho de Samuel [32, 33] tornou-se um marco, o qual consistia de um programa que aprendia a jogar damas, baseado nas experiências obtidas em jogos anteriores. O seu programa continha uma série de parâmetros, cada qual com um valor numérico. Estes valores numéricos eram ajustados pela experiência (conjunto de exemplos) considerando a importância do mesmo para um determinado movimento (decisão),

(ii) aprendizagem **simbólica** de conceitos.

Os conceitos descrevem classes de eventos, objectos ou relações entre eles. Uma das formas importantes de relacionar conceitos é a de fazer qualquer afirmação sobre eles, por exemplo: o rato roeu a roupa do rei. Esta afirmação é chamada de proposição, a qual apresenta uma afirmação que relaciona um sujeito (o rato) e um predicado (roeu). O sujeito e o predicado correspondem respectivamente: ao item em relação ao qual a afirmação é feita, e o que é afirmado em relação ao sujeito. Independente da veracidade (verdadeiro ou falso) da proposição, são essas proposições que ligam (associam) os elementos mentais. Este paradigma orienta a aquisição de conceitos e conhecimento estruturado.

A aprendizagem simbólica tem como uma das bases os modelos humanos de aprendizagem cognitiva, os quais podem utilizar a lógica ou grafos no lugar de métodos numéricos ou estatísticos. O trabalho de Buchanan ("*Meta-Dendral*") [34] o qual gera regras para uso no

sistema pericial "*Dendral*" [35] e o trabalho de Wiston [36] em sistemas de aprendizagem estrutural, são exemplos pioneiros deste paradigma na aprendizagem computacional, e

(iii) **outros métodos** de aprendizagem e sistemas de aprendizagem com **conhecimento intensivo**.

A partir de meados da década de setenta e considerando os paradigmas anteriores, os investigadores expandiram os seus interesses para além da aprendizagem de conceitos isolados a partir de exemplos. Passaram a investigar uma vasta gama de métodos de aprendizagem (por instrução, por analogia, de descoberta de conceitos e classificação, etc.) e métodos baseados em conhecimento intensivo (métodos de aprendizagem em sistemas os quais a partida possuem uma quantidade expressiva de conhecimento).

Estando ainda em aberto uma definição que possa cobrir amplamente o significado de aprendizagem (relativa as capacidades apresentadas pelos animais), inúmeras aproximações têm contribuído para o enquadramento nesta disciplina de características e propriedades a ela pertencentes. Das referências, abaixo mencionadas, relativas as definições de aprendizagem, extraem-se as seguintes características;

de Simon [37]:

(i) denota mudanças num sistema,

(ii) estas mudanças capacitam o sistema a executar, numa próxima vez, a mesma actividade ou actividade da mesma população de forma mais eficiente,

de Minsky [38]:

(i) denota mudanças úteis na nossa mente,

de Michalski [39]:

(i) é a construção e a modificação da representação do que está sendo experimentado, e

de Carbonell [40]:

(i) é a habilidade de executar novas tarefas, as quais não eram possíveis anteriormente,

(ii) é melhorar (maior precisão, menor tempo, etc.) a execução de antigas tarefas.

Das características apresentadas por estas definições, e analisando-as na perspectiva que a aprendizagem está associada à mudança dos estados mentais de um aprendiz (agente), dois tipos de aprendizagem automática apresentam-se:

(i) a aquisição de conhecimento, e

(ii) o refinamento de habilidades.

A primeira, e de interesse para esta dissertação, diz respeito à aquisição de novas informações simbólicas acopladas com a habilidade de aplicá-las de maneira útil. A segunda, refere-se à prática repetida e à correcção dos desvios do comportamento desejado, e relaciona-se ao conhecimento chamado de sub-simbólico (conhecimento adquirido, por exemplo pelas redes neuronais, o qual não é de fácil

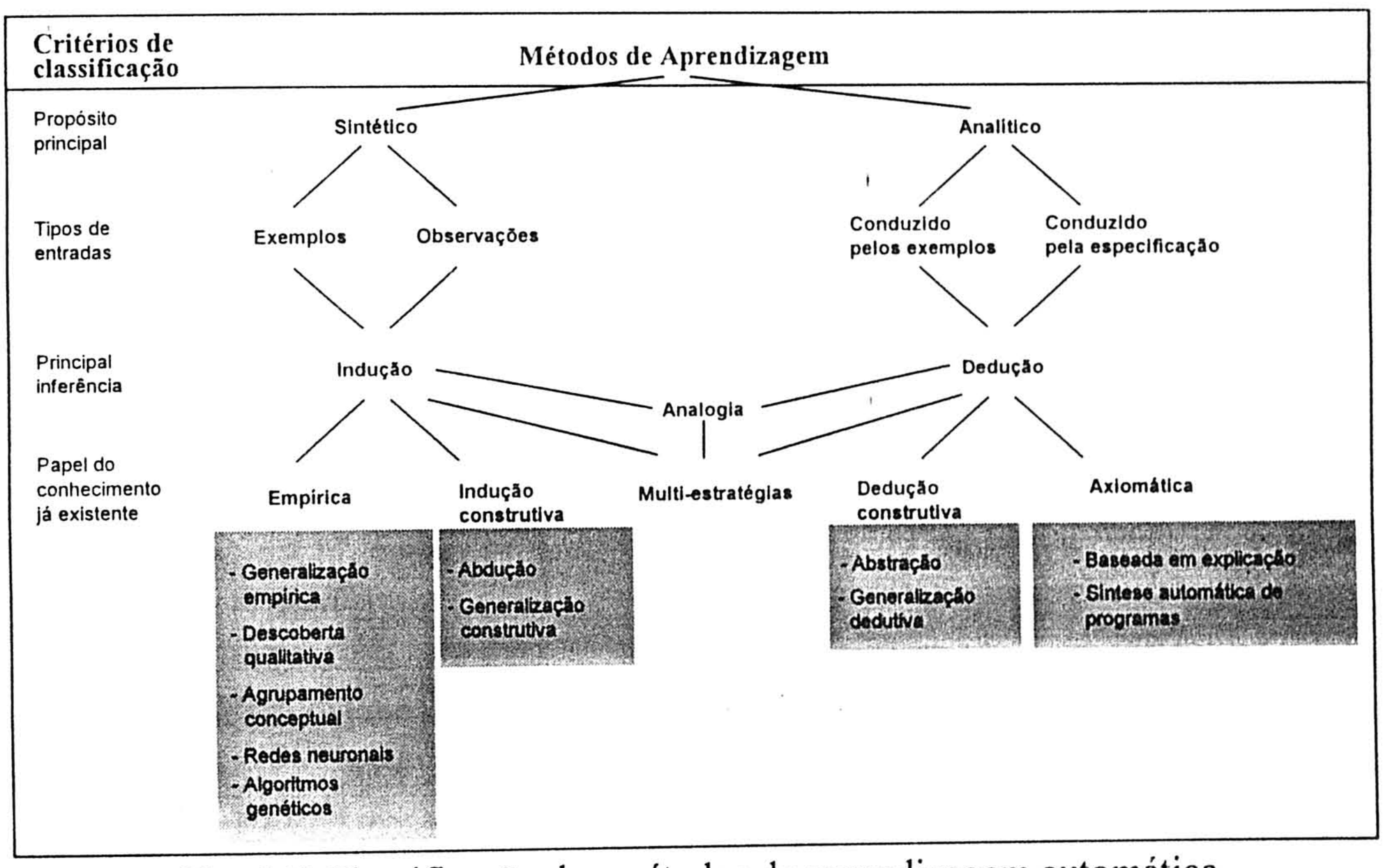
entendimento pelo ser humano), cujo processo de geração é encontrado fundamentalmente nos sistemas adaptativos.

1.3.2.2.1 Classificações

Os processos de aprendizagem podem ser classificados de diferentes pontos de vista, como as classificações apresentadas em [41, 42]. Para a compreensão geral desses métodos, seleccionamos a classificação [fig. 14] encontrada em [43], a qual utiliza os critérios de:

(i) propósito da aprendizagem,
(ii) tipos de entrada da informação para o processo de aprendizagem,
(iii) inferência utilizada para geração de novo conhecimento, e
(iv) o papel do conhecimento já existente, no processo de geração de conhecimento,

para o enquadramento dos mesmos.



[fig. 14] Classificação dos métodos de aprendizagem automática

Considerando os critérios utilizados para a classificação dos métodos de aprendizagem temos:

(i) **o critério de propósito principal** ou seja o objectivo principal dos métodos de aprendizagem. Este critério conduz à divisão dos métodos em **sintéticos** e **analíticos**.

Os sintéticos visam fundamentalmente a criação de novo ou melhor (de acordo com algum objectivo) conhecimento.

Os analíticos estão interessados na reformulação (transformação ou organização) de um conhecimento existente num melhor (de acordo com algum objectivo),

(ii) **o critério dos tipos de entrada**. Este critério agrupa os métodos de aprendizagem quanto ao tipo da entrada de informação para estes.

A aprendizagem é efectivada a partir de **exemplos** (positivos ou negativos), previamente classificados, quando o conhecimento fonte independentemente da sua origem (professor, especialista, modelo simulado, etc.), exemplifica um objecto ou evento.

É efectivada a partir de **observações**, quando as entradas necessitam ser estruturadas pelo aprendiz, uma vez que as mesmas não se encontram previamente classificadas.

É **conduzida pelos exemplos**, quando a reformulação do conhecimento já existente consiste em explicar o exemplo utilizando a teoria existente, gerando, generalizando ou especializando conhecimentos. Um exemplo é o conceito de suicídio, o qual é um caso particular da teoria dos assassinatos, e pode ser deduzido a partir da teoria existente.

É **conduzida pela especificação** quando o conhecimento existente orienta quais as associações que devem ser mantidas entre as entradas e as saídas para executar um procedimento. Estão incluídas nesta classe a síntese automática de programas [44] (processo de transformar a especificação de um programa num algoritmo executável) e o aconselhamento,

(iii) **o critério do principal método de inferência empregue**. Este critério classifica os métodos de aprendizagem quanto à forma de raciocínio empregue na transformação dos estados mentais.

É **indutivo** quando a partir de dadas consequências gera premissas hipotéticas.

É **dedutivo** quando deriva consequências a partir de dadas premissas. Se considerarmos a proposição C a qual pode ser demonstrada por P e BK, representada por:

$P \& BK \mid\!> C$	*onde:* P *é uma premissa,* BK *é o conhecimento existente num aprendiz, e* C *é uma consequência*

Se o raciocínio for: observar P, o qual é verdadeiro, e utilizar o conhecimento BK, o aprendiz pode afirmar que C é verdadeiro. Estamos então diante de uma **dedução**. Esta dedução apresenta uma forma fraca caso P seja parcialmente verificado ou BK não seja totalmente correcto.

Se o raciocínio for: observar C e por hipótese gerar P como explicação ou generalização da observação, então estamos face a uma **indução**. Neste caso, como o número de hipóteses pode ser muito grande (é em geral infinito), BK é visto como um orientador, possibilitando o decréscimo quantitativo das hipóteses.

Ainda no critério do principal método de inferência empregue, encontra-se a inferência por **analogia**, a qual combina tanto a indutiva como a dedutiva. A analogia envolve a transferência de certas propriedades de uma base de conhecimento (um conceito, um procedimento, etc.) para um conhecimento alvo. A inferência dedutiva é empregue para reconhecer ou descobrir uma analogia, sendo necessário detectar uma semelhança ou relações comuns entre a base de conhecimento e o conhecimento alvo e então, gerar hipóteses para além das relações comparadas. Uma vez determinadas as relações comuns, a transferência de propriedades de uma base de conhecimento para o conhecimento alvo é executada através da inferência dedutiva. Em aprendizagem automática a analogia tem sido utilizada na aprendizagem de conceitos [45, 46] e na solução de problemas [47], e

(iv) **o critério do papel do conhecimento já existente**. O método é **empírico** quando utiliza conhecimento já existente de uma maneira não intensiva e com inferência indutiva. Os trabalhos mais comuns nesta área repousam na geração de descrições a partir de exemplos, utilizando basicamente atributos que estão presentes na descrição dos exemplos.

A **indução construtiva** por sua vez, para além de utilizar a inferência indutiva, baseia-se num uso intensivo do conhecimento existente para a criação de conceitos ou explicações.

A **abdução** como um caso da indução construtiva utiliza o conhecimento existente para orientar a preferência para uma hipótese mais plausível de entre as possíveis. A formulação da abdução é dada por: "dado um consequente `C` e um conhecimento já existente `BK`, o aprendiz busca por uma premissa `P` (hipótese) tal que `P&BK` fortemente ou fracamente conclua `C`".

`P&BK|>C`	*onde:* `P` *é a premissa,* `BK` *é o conhecimento existente, e* `C` *é a consequência*

A **dedução construtiva** é um processo baseado em conhecimento, o qual consiste na transformação das descrições de um espaço de representação ou linguagem para outro, preservando as informações importantes para o objectivo. Na dedução construtiva encontramos a **abstracção** e a **generalização dedutiva**.

A abstracção reduz o total de informação contida no conhecimento, porém preserva (implicitamente ou explicitamente) a informação necessária para o objectivo. A abstracção geralmente troca o espaço de representação ou linguagem.

A generalização dedutiva está voltada para construir generalizações que são deduções lógicas da base de conhecimento e geralmente utiliza o mesmo espaço de descrição ou linguagem..

O método **axiomático** baseia-se no processo de dedução puro, utilizando um completo e consistente conhecimento já existente. Este conhecimento desempenha um papel análogo aos axiomas nas teorias formais.

A diferença entre a dedução construtiva e a axiomática dedutiva está no grau de liberdade das alterações do espaço ou da linguagem utilizada para a representação do conhecimento. A primeira admite uma variedade de formas de transformação do conhecimento (essencialmente é preservadora da verdade, porém pode utilizar a inferência indutiva e o raciocínio aproximado), a segunda está baseada nos métodos formais da lógica.

Por fim a aprendizagem **multi-estratégia** está baseada na utilização combinada de estratégias/paradigmas da aprendizagem. Por exemplo a integração da generalização empírica e da aprendizagem baseada em explicações ou a combinação de métodos simbólico e subsimbólico [48]. Sistemas baseados em aprendizagem multi-estratégia, tornam-se capazes de adaptarem a estratégia de aprendizagem em função do tipo de tarefa e cooperarem na resolução de problemas envolvendo vários tipos de imperfeições (quer sejam do conhecimento já existente ou da entrada da informação).

O anexo F apresenta a aprendizagem simbólica automática por indução empírica e por dedução axiomática. A aprendizagem simbólica por indução empírica suporta o enquadramento da aprendizagem tratada por esta dissertação.

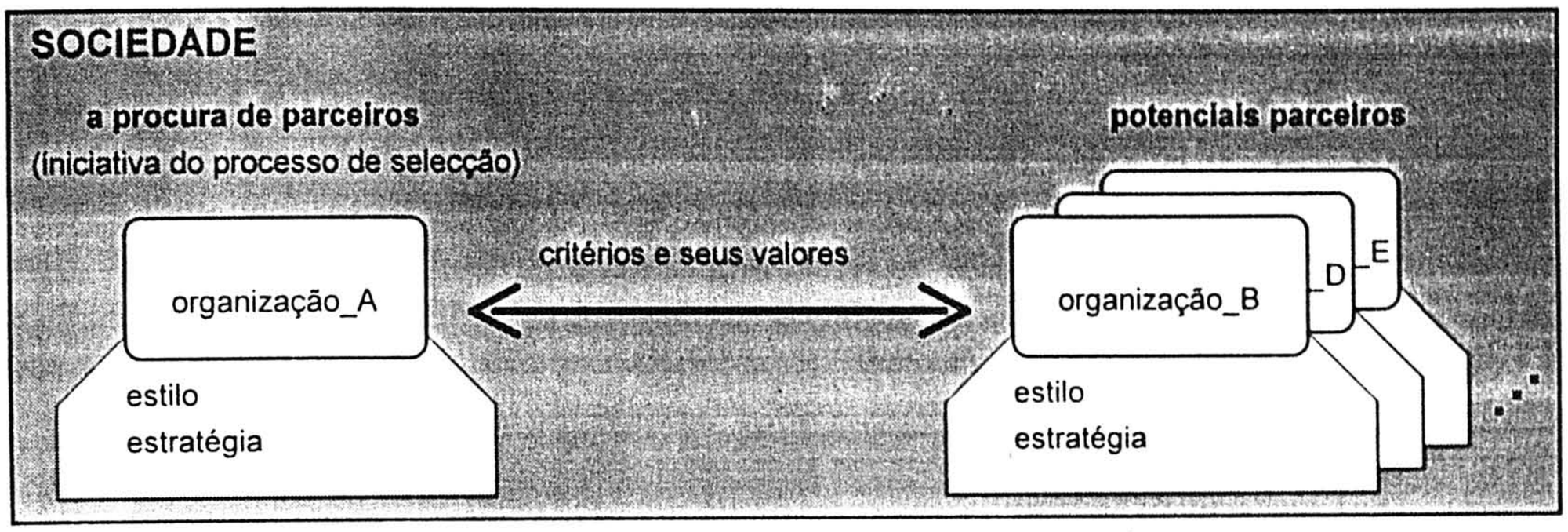
1.3.2.3 Aprendizagem em Sistemas Multiagente

Num sistema inerentemente distribuído, como é o caso de uma comunidade de organizações e de cada uma das organizações de *per se*, a distribuição da informação é uma das características desta classe de sistemas. As soluções dos mais diversos problemas ou a melhoria na maneira de implementar as soluções, requerem que haja disponibilidade destas informações por parte dos membros da comunidade. Todavia, e em muitos casos, as mesmas não se encontram explicitadas ou não são possíveis de serem obtidas de maneira directa (i.e. um pedido de informação e uma resposta afirmativa a este pedido com conteúdo consistente), havendo a necessidade de recorrer à inferência indutiva ou dedutiva para poder obtê-las.

Como exemplo do acima exposto, e como resposta à questão do porquê a aprendizagem é útil em sistemas distribuídos, vejamos o caso de uma organização, inserida numa sociedade de organizações, que pretenda seleccionar parceiros para colaborarem no alcance de um certo objectivo (por exemplo o fornecimento de um bem/serviço) [fig. 15]. O processo de selecção é baseado em vários critérios, por exemplo:

(i) critérios no domínio organizacional (tempo de actuação da organização no mercado, etc.),
(ii) critérios no domínio técnico (dimensões do produto a ser fornecido; etc.), e
(iii) critérios no domínio comercial (prazo de pagamento, etc.),

através dos quais a organização que tomou a iniciativa do processo de selecção, procederá à avaliação e escolha dos potenciais parceiros.



[fig. 15] Organização em busca de parceiros

Cada critério pode possuir vários valores considerados válidos. Um desses valores, em cada um dos critérios, é referido como um valor de satisfação, podendo, se existirem, os demais serem:

(i) valores mais vantajosos (maior satisfação), ou
(ii) menos vantajosos (menor satisfação).

Se tomarmos como exemplo o critério "prazo de pagamento", pertencente ao domínio comercial, com os valores válidos "30, **60**, 90 e 120 dias", e referirmos a "60 dias" como o valor de satisfação, o valor menos vantajoso será "30 dias" e os mais vantajosos serão "90 e 120 dias".

A existência de vários valores para cada um dos critérios, possibilita a troca dos mesmos, caso os potenciais parceiros não possam atendê-los no valor referido como de satisfação.

Além dos valores válidos, cada um dos critérios possui uma utilidade que caracteriza a importância (do critério e do seus valores) para a organização.

Para proceder às trocas, i.e. gerar valores para os critérios os quais atendam o conceito de utilidade, as organizações envolvidas, necessitam ser orientadas por conhecimentos que lhes permitam alcançar o objectivo do processo de selecção (contratar parceiros ou serem os parceiros contratados). Estes conhecimentos, fundamentalmente, são:

(i) o perfil (estilo) da organização,
(ii) a posição da organização no mercado,
(iii) os planos da organização,
(iv) o que a organização conhece sobre o mercado, e
(v) o que a organização conhece sobre os potenciais parceiros.

Estes conhecimentos orientam, os comportamentos das organizações, na execução de uma actividade (variando de cooperativo a impositivo), os quais são

implementados por estratégias correspondentes, neste exemplo, de como proceder para contratar.

Grande parte do conhecimento (por exemplo: o que uma organização conhece sobre a sociedade e sobre os potenciais parceiros) que compõe um estilo e tendo como consequencia o emprego de estratégias adequadas (por exemplo: satisfação do objectivo através da racionalização de recursos), não é conseguido através de perguntas directas (de uma organização para outra) e objectivas (qual é seu estilo? ou qual é a estratégia que utiliza num processo de selecção?). No entanto, este conhecimento pode ser inferido (com maior ou menor grau de exactidão) através das metodologias de aprendizagem baseadas em experiências.

Um dos principais objectivos da Aprendizagem Automática em Sistemas Multiagente ("*Multiagent Learning - MAL*") é o de melhorar o desempenho de Sistemas baseados em Inteligência Artificial Distribuída. Devido as constantes interacções e o partilhamento de conhecimento [49] entre agentes de uma comunidade, a aprendizagem em sistemas resolvedores de problemas com múltiplos agentes contribui para o refinamento do comportamento social desses agentes.

Em sistemas multiagente, dois tipos de aprendizagem podem ocorrer [50]:

(i) **os agentes aprendem com a comunidade**

Como uma comunidade, a aprendizagem dos agentes ocorre a nível da coordenação e a nível da designação de tarefas e recursos. Ambas são possíveis através do partilhamento da informação e do conhecimento. Por exemplo a coordenação pode ser melhorada quando é aprendido:

(i.i) qual a especialização dos agentes na comunidade. O agente "organização_A" sempre oferece valores do "critério_T" que satisfazem, ou

(i.ii) em termos de característica. O agente "organização_C" e o agente "organização_Y" comportam-se de maneira cooperativa.

Do ponto de vista da tarefa, a aprendizagem dos agentes ocorre no comportamento adoptado para a geração dos valores dos critérios, por exemplo: para obter valores mais que satisfatórios para o "critério_V", relaxam-se o valor do "critério_B" e do "critério_G", e

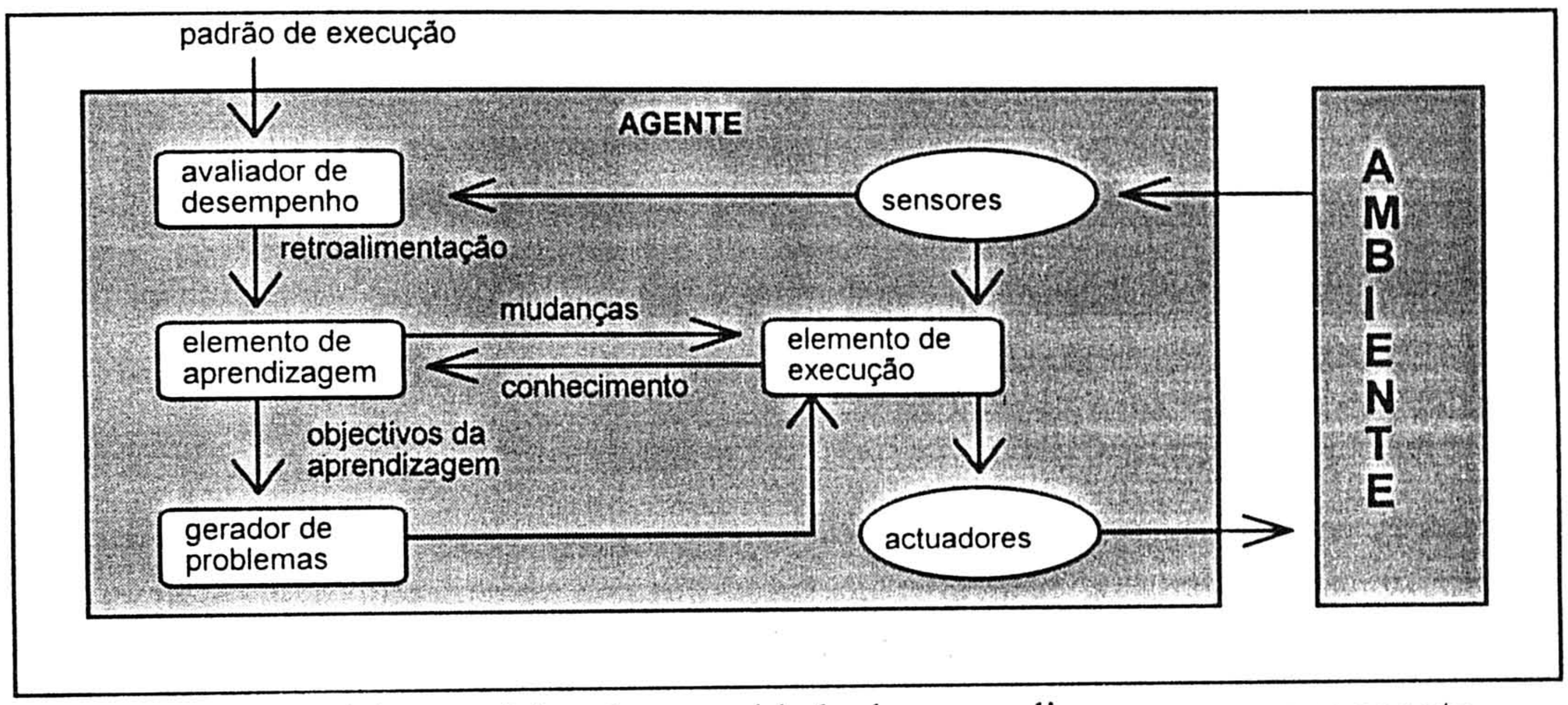
(ii) **cada agente aprende por si através do ajuste de suas acções**

Esta ocorrência da aprendizagem está direccionada para a melhoria do perfil do agente quando da resolução de problemas. Ela pode ocorrer ou através da revisão de suas heurísticas e crenças ou através da observação de como os demais agentes da comunidade agem face a uma certa questão.

1.3.2.3.1 Arquitectura

Quanto à arquitectura de um agente dotado da capacidade de aprendizagem, o modelo genérico apresentado em [51] descreve-a como contendo quatro componentes conceituais [fig. 16]:

(i) um avaliador de desempenho,
(ii) um elemento de aprendizagem,
(iii) um elemento de execução, e
(iv) um gerador de problemas.



[fig. 16] Modelo genérico da capacidade de aprendizagem para um agente

As funções destes componentes são as seguintes:

(i) **o avaliador de desempenho** é o responsável por criticar o desempenho do agente em relação à execução das suas actividades, comunicando ao elemento de aprendizagem um valor que expresse este desempenho.

A percepção de um estado, por si só, não indica o desempenho. Há necessidade de existir um padrão, o qual é externo ao agente, que estabeleça esta medida. Por exemplo num jogo de xadrez, o agente pode receber a informação de que colocou seu oponente no estado de cheque-mate. Porém, é necessário, que exista um padrão que indique se este estado é satisfatório ou não,

(ii) **o elemento de aprendizagem** é o responsável por fazer as melhorias (mudanças) nas acções que serão executadas pelo elemento de execução. É no elemento de aprendizagem que residem os algoritmos de aprendizagem,

(iii) **o elemento de execução** é o responsável pela selecção das acções a serem executadas. Este elemento possui a capacidade de perceber o ambiente e decidir as acções a serem executadas, e

(iv) **o gerador de problemas** é o responsável por sugerir acções que conduzam a novas e expressivas experiências. Se o agente é desejoso de explorar novas acções em curtas execuções, é possível descobrir

melhores acções para longas execuções. A geração de tarefas é utilizada para sugerir estas acções exploratórias e experimentais.

Concretizando este modelo, vamos instanciá-lo utilizando o caso citado da organização interessada na contratação de parceiros para colaborarem no alcance de um certo objectivo (por exemplo o fornecimento de um bem/serviço). Então, nesta situação teríamos que:

(i) **o elemento de execução** consistiria de uma colecção de conhecimentos sobre:

(i.i) o perfil da organização, a posição da organização no mercado, os planos da organização, o que a organização conhece sobre o mercado e o que a organização conhece sobre os potenciais parceiros, etc., e

(i.ii) os procedimentos baseados no estilo e nas estratégias para implementar a geração dos valores dos critérios para o processo de selecção.

O elemento de execução é o responsável pelo envio destes valores às organizações (agentes) com as quais está em contacto,

(ii) **o elemento de aprendizagem** formularia os objectivos do que aprender, por exemplo: aprender o comportamento do potencial parceiro quando um determinado estado dos critérios apresenta certos valores,

(iii) **o avaliador de desempenho** observaria o comportamento dos potenciais parceiros e enviaria para o elemento de aprendizagem, informações sobre o desempenho do processo de selecção, por exemplo: se está a ocorrer convergência para os valores desejados ou distanciamento destes (por valores mais ou menos vantajosos),

(iv) diante do valor da medida de avaliação, **o elemento de aprendizagem** formularia um novo procedimento, por exemplo:

(iv.i) buscar a convergência aos valores desejados através do relaxamento nos critérios de menor utilidade, ou

(iv.ii) o distanciamento destes por valores mais vantajosos, nos critérios de maior utilidade,

e enviaria ao elemento de execução este conhecimento (regras de produção, redes semânticas, roteiros, etc.), e

(v) ocasionalmente, **o gerador de problemas** poderia experimentar (simular) o que aconteceria se determinado valor de um critério fosse enviado, dado o comportamento de um parceiro o qual foi aprendido em experiências anteriores.

Um modelo de um sistema multiagente dotado com a capacidade de aprendizagem é apresentado no anexo G. É a arquitectura do sistema "*MALE - Multi-Agent Learning Environment*" [52], para aprendizagem num ambiente de agentes cooperativos.

1.3.2.3.2 Conclusão

Nesta secção enunciamos vários paradigmas que informam as capacidades de aprendizagem de sistemas computacionais.

Formalizamos também as possibilidades de aprendizagem em Sistemas Multiagente ilustrando, a partir de uma arquitectura conceptual para um agente cognitivo, situações em que seria importante os agentes estarem dotados de capacidades de evolução (aprendizagem). Pensamos também ter tornado claro o salto qualitativo que seria conseguido pela (pequena que seja) introdução de tais facilidades na implementação dos Sistemas Multiagente como é, parcialmente, um dos objectivos desta tese.

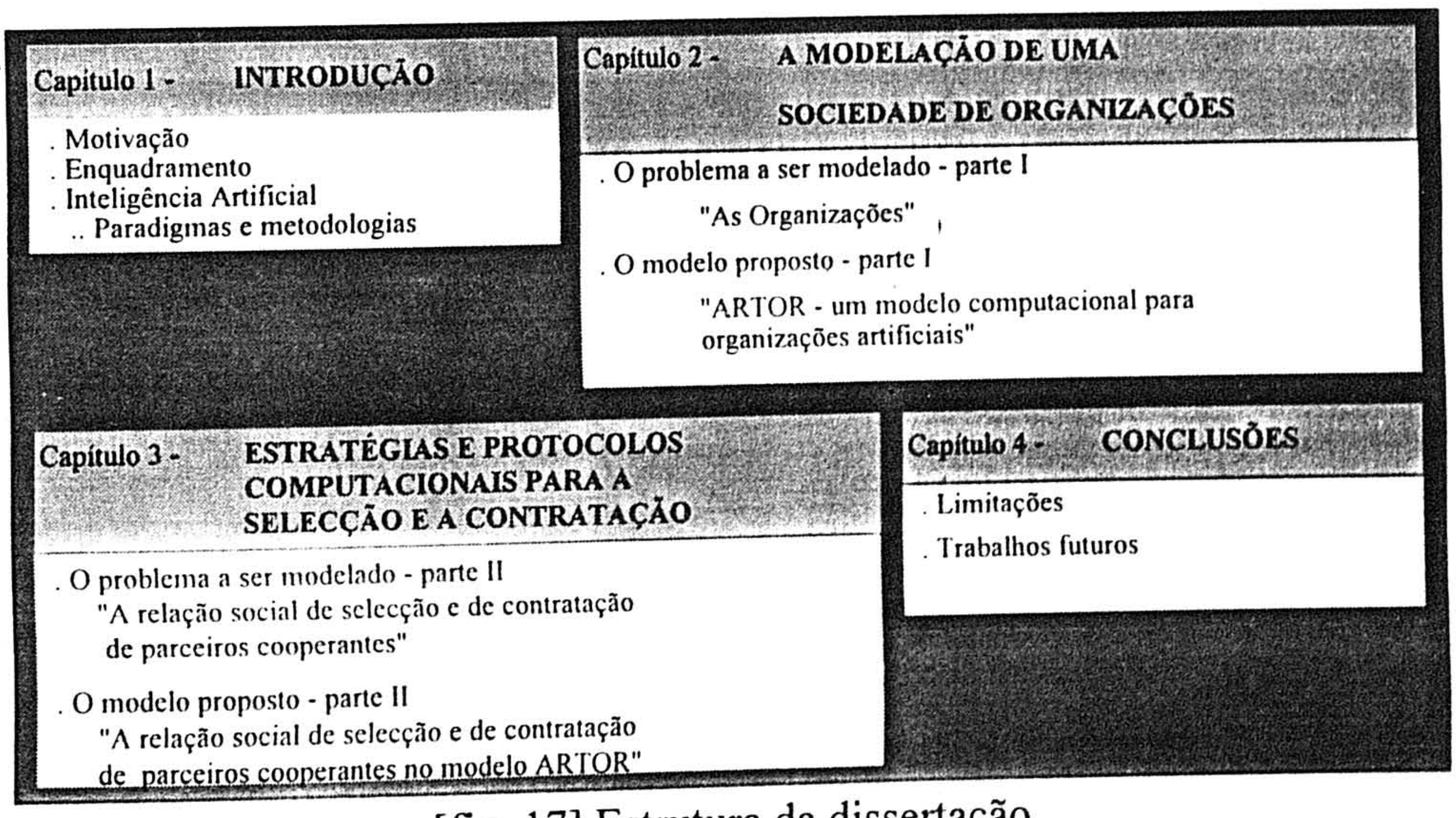
1.4 A Organização da Tese

Com o objectivo de fornecer uma visão geral do conteúdo e da organização dessa dissertação a [fig. 17] apresenta de forma sucinta a estrutura da mesma.

O Capítulo 2 contém a primeira parte do problema do mundo real a ser modelado através de metodologias computacionais, mais especificamente os Sistemas Multiagente e a Aprendizagem Simbólica Automática. Inicialmente são descritas as organizações, fornecendo uma visão sobre as mesmas numa sociedade de organizações, através:

(i) de seus componentes,
(ii) dos papeis e estruturas desses componentes, e
(iii) das relações intra e inter-organizacionais.

Em seguida, é descrito o modelo "*ARTOR*" ("*ARTificial ORganizations*"), como um sistema computacional baseado em agentes, como resposta ao problema estudado. A descrição ocorrerá através da apresentação da arquitectura dos agentes representativos das organizações, suas capacidades, conhecimentos e a comunicação intra e inter-organizações (agentes). Por fim apresentam-se outros modelos computacionais para organizações também baseados em agentes, concluindo com uma comparação entre o modelo proposto e os analisados.



[fig. 17] Estrutura da dissertação

O Capítulo 3 apresenta a segunda parte do problema do mundo real a ser modelado, também baseado nas metodologias acima citadas. Inicia desenvolvendo o problema específico de uma relação social (inter-organizações) de contratação de parceiros para cooperarem no alcance de um objectivo, através de um processo de selecção. Esta relação fornece suporte a uma das etapas do processo de terceirização das organizações ("*outsourcing*") e da criação de organizações virtuais. Em seguida são descritas, em termos computacionais, as etapas do processo de selecção e de contratação (o convite, a avaliação, a detecção de situações de conflito, a negociação como metodologia para resolução desses conflitos, a contratação e a aprendizagem). Por fim são apresentadas outras metodologias de contratação, negociação e aprendizagem em relações similares, concluindo com uma comparação e reflexão sobre a metodologia proposta e as demais analisadas.

O Capítulo 4 contém as conclusões desta dissertação, apresentando as limitações dos modelos propostos (parte I e parte II). Finaliza com a citação dos trabalhos futuros, utilizando o ambiente "*ARTOR*" para a implementação de outras relações sociais inter e intra-organizações.

1.5 Principais Contributos e Potenciais Áreas de Aplicação

1.5.1 Introdução

Os primeiros investimentos nas organizações, para o tratamento da informação através de recursos computacionais, ocorreram nos sistemas de informação das áreas operacionais e administrativas. Estes sistemas caracterizavam-se pelo tratamento dos dados e processos de forma isolada i.e. cada sistema de informação possuía sua base de dados com baixo nível de integração. Gradualmente passaram a desenvolver-se sistemas de informação para o suporte de gestão, os quais a partir dos dados gerados pelas bases operacionais e administrativas, permitiam extrair sínteses que auxiliavam a gestão das organizações através de posições consolidadas das suas actividades. Como a integração, a integridade e a disponibilidade das informações encontravam-se ainda comprometidas, as organizações passaram a procurar modelos que possibilitassem estruturar o tratamento da informação de forma corporativa e em tempo real.

Modelos corporativos são modelos que representam os dados tratados pelas organizações de forma integrada e partilhada pelos múltiplos processos que as compõem, por exemplo: os dados que representam um cliente numa aplicação de poupança numa instituição bancária, tais como nome, endereço, etc, são os mesmos, em termos de representação e armazenamento em recursos computacionais, quando este mesmo cliente tem outras operações de investimentos ou empréstimos, na mesma organização. Um outro exemplo é o da rotina para a venda de um produto estar interligada com as rotinas: de controlo do armazenamento, de reposição do produto vendido, de contabilização, de fluxo de caixa, de planeamento, etc.

Os modelos corporativos foram, inicialmente, suportados pelos modelos hierárquicos, seguido dos modelos em rede e por fim pelos modelos relacionais (E-R), os quais através de metodologias e de "*softwares*" ("*ADABAS*", "*ORACLE*", "*DB2*",

etc.) possibilitaram um avanço significativo de implementação. Acompanhando estas metodologias, a Engenharia de "*Software*" e o avanço na área da Inteligência Artificial permitiu o desenvolvimento de Sistemas de Informação mais robustos, de Sistemas Periciais [53] e de Sistemas de Aprendizagem Automática aplicados a problemas do mundo real.

Algumas das áreas emergentes do tratamento da informação para:

(i) os mercados electrónicos,

(ii) a economia global,

(iii) a reengenharia dos processos e das organizações,

(iv) os processos automatizados inteligentes de tomada de decisão,

(v) a preservação do conhecimento, o qual é um dos principais itens do património das organizações [153], e

(vi) a descoberta de novos conhecimentos [114, 115] nas bases de dados corporativas,

têm motivado a comunidade de investigadores, de técnicos e de empresários a procurarem soluções que permitam acrescer qualitativamente os processos que suportam essas áreas, a fim de atender a competitividade nos mercados.

1.5.2 Contributos

Uma **organização** é um conjunto de **unidades funcionais** (operacionais e administrativas) relacionadas segundo uma estrutura que permite, através do desempenho das actividades (definidas por objectivos parciais, os quais são produtos da decomposição dos objectivos organizacionais globais), alcançarem os objectivos globais da organização. Estas **unidades funcionais** podem ser vistas como **centros cognitivos**, dotados de percepção, raciocínio e memória o que faculta a comunicação, o planeamento, a coordenação, a execução de suas especialidades e a aprendizagem. A organização como um todo pode então ser considerada como um conjunto composto de: centros cognitivos e de uma base de conhecimento corporativa.

De forma geral o contributo desta dissertação recai numa arquitectura baseada em Sistemas Multiagente e Aprendizagem Simbólica Automática que possibilita o modelação (de, pelo menos, parte) das organizações segundo a perspectiva acima apresentada.

De forma específica o contributo recai na modelação de uma relação social interorganizacional: a de contratação de organizações que estejam interessadas em cooperarem no alcance de um objectivo.

Os principais contributos são:

(i) a implementação de uma arquitectura que possibilita modelar, baseada em agentes, as unidades funcionais das organizações,

(ii) a implementação de uma metodologia de coordenação de actividades distribuídas pelas unidades funcionais da organização. Esta metodologia compreende:

(ii.i) a representação do conhecimento para os objectivos e planos organizacionais,

(ii.ii) a distribuição ou redistribuição (caso a unidade não tenha disponibilidade para executa-la) das actividades pelas unidades funcionais, e

(ii.iii) o controlo das actividades,

(iii) o desenvolvimento de uma metodologia para automação da relação social de contratação a qual inclui:

(iii.i) a representação do conhecimento para os critérios de selecção,

(iii.ii) a coordenação da actividade de selecção,

(iii.iii) o protocolo de negociação com vista à satisfação dos critérios, e

(iii.iv) a representação do conhecimento para os estilos, estratégias e tácticas com o qual o agente orientará as negociações, e

(iv) a criação, por cada um dos agentes de uma sociedade de organizações, do modelo de conhecimento (no domínio da contratação):

(iv.i) das organizações participantes no processo de selecção, e

(iv.ii) da sociedade,

através de aprendizagem automática.

1.5.3 Aplicações

A utilidade dos assuntos tratados por este trabalho apresenta-se potencialmente nas seguintes aplicações:

(i) para incrementar a automação e a integração das informações nas organizações, dando um passo para além das bases corporativas,

(ii) para a automação de processos de contratação, em organizações:

(ii.i) que necessitam de licitações para a aquisição de bens/serviços,

(ii.ii) que queiram automatizar a tomada de decisão no domínio da selecção,

(ii.iii) que queiram ter uma constituição virtual, e

(ii.iv) que participem dos mercados electrónicos.

(iii) para a preservação do conhecimento aprendido no domínio, e

(iv) como auxiliar nos processos de análise e selecção de dados "*data mining*", uma vez que a informação está distribuída e parcialmente classificada.

1.6 Conclusões

A área de Inteligência Artificial e as suas componentes Inteligência Artificial Distribuída - Sistemas Multiagente e Aprendizagem Simbólica Automática, apresentam paradigmas que suportam a modelação de problemas inerentemente

distribuídos do mundo real, tal como é o caso das organizações numa sociedade de organizações.

As organizações são constituídas por entidades distribuídas, tanto em termos de funcionalidades quanto em termos de espaço. Podemos olhar para cada uma das suas unidades funcionais, como uma unidade cognitiva capaz de executar actividades que concorram para o alcance do objectivo global. Esta cognição sustenta a possibilidade de tomar decisões, executar tarefas dentro da especialidade e aprender.

Pensar num modelo para as organizações como se fossem sistemas monolíticos seria, além de negar a distribuição, característica que lhe é própria, atribuir-lhes um tamanho tal que comprometeria a fiabilidade, a compreensão e a flexibilidade, além de dotá-los de uma grande complexidade.

Modelá-las como se fossem sistemas distribuídos de controlo centralizado seria conduzi-las a um paradigma que traria problemas devido a elevada concentração de decisões a serem tomadas por um único componente. Para além disso, iríamos descaracterizá-las quanto à autonomia, (mesmo que no mundo real esta autonomia seja parcial), que cada unidade funcional possui. A modelação deste problema através de metodologias de Sistemas Multiagente e de Aprendizagem Simbólica Automática, prova ser adequada na medida em que:

(i) se aproxima do modo como na realidade este problema é formulado,
(ii) diminui a complexidade dos componentes, uma vez que cada unidade funcional constitui um módulo,
(iii) aumenta a flexibilidade de alterações, inclusões, eliminações e movimentações das unidades funcionais,
(iv) permite uma maior eficiência, usufruindo do partilhamento do tempo através do paralelismo,
(v) permite um tratamento a falhas quando existe mais de uma unidade funcional com capacidade de executar a tarefa,
(vi) permite o balanceamento da carga computacional através da distribuição dos componentes de "*software*" [54],
(vii) baseada em Aprendizagem Simbólica Automática, cada unidade funcional passa a usufruir das experiências proporcionadas pela execução das actividades de sua competência, através da aprendizagem empírica, e contribui para incremento do conhecimento corporativo.

Actualmente a Reengenharia das Organizações tem tomado lugar de destaque na reorganização dos sistemas e processos. Esta reengenharia está fundamentada na reinvenção dos processos organizacionais, sendo a tecnologia baseada em recursos computacionais e a aprendizagem, duas das componentes que desempenham um papel relevante neste repensar organizacional. Parafraseando Hammer e Champy [55]:

> "Muitos executivos e gerentes sabem como proceder para pensar dedutivamente. Isto é, eles são bons na definição dos problemas, na procura e na validação de diferentes soluções. Mas aplicar tecnologia da informação para a Reengenharia de Negócios requer raciocínio indutivo - a habilidade de primeiro reconhecer a importância da

solução e então procurar o problema que pode ser resolvido com ela. Estes problemas, provavelmente as organizações não sabem que eles existem. A questão que deve ser colocada é: "Como nós (as organizações) podemos usar a tecnologia de forma que nos permita fazer coisas que nós ainda não fazemos?".”

Capítulo 2

A MODELAÇÃO DE UMA SOCIEDADE DE ORGANIZAÇÕES

2.1 Introdução

A maioria das actividades orientadas para a produção de bens e serviços são planeadas, coordenadas e executadas intra ou inter-organizações. As organizações são entidades de diferentes tamanhos, características, estruturas e objectivos. Existem organizações com fins lucrativos as quais são chamadas de empresas e as com fins não lucrativas como é o caso da Igreja, Forças Armadas, serviços públicos, etc.

Independentemente da sua constituição, uma organização pode ser vista como uma interface entre um ambiente interior ("*inner*") e um ambiente exterior ("*outer*"). A efectivação desta interface ocorre através das relações estabelecidas entre esses dois ambientes. As relações podem ser do tipo passiva (observação) ou do tipo activa (solicitação ou envio de informação, matéria ou energia de maneira espontânea ou por solicitação). Estas relações conduzem ao alcance dos objectivos da organização e à satisfação dos demais componentes do mundo exterior, gerando novos estados no ambiente interno e externo.

O ambiente interno de uma organização é composto por:

(i) recursos humanos,

(ii) recursos não humanos (oriundos do mundo natural como é o caso de certas matérias-primas, e do mundo artificial como por exemplo: capital, tecnologia),

(iii) uma estrutura física (local onde os recursos são dispostos e operam),

(iv) uma estrutura conceptual (conceitos que orientam os comportamentos e as acções),

(v) um conjunto de "tarefas-meio" (tarefas que suportam o alcance dos objectivos, porém não actuam directamente sobre eles, por exemplo: a manutenção do local onde os recursos estão dispostos e são operados). Estas tarefas são, ou rotinas (activadas em função de determinado espaço de estados) ou não rotinas ("*adhoc*"), e

(vi) um conjunto de "tarefas-fins" (tarefas que suportam directamente o alcance dos objectivos, por exemplo: encaixar peça A na peça B para o fabrico do produto P1. O produto P1 é um produto produzido pela organização). Estas tarefas são, ou rotinas (activadas em função de determinado espaço de estados) ou não rotinas ("*adhoc*").

Para que todos estes componentes do ambiente interno possam operar ou serem operados eficiente e eficazmente e, para que as adaptações geradas pelas relações com o ambiente externo possam ocorrer, o ambiente interno das organizações necessita ser gerido. A gestão ocorre através das acções de planear, estruturar fisica e conceptualmente (comportamentos desejados) e coordenar as dependências entre acções e recursos.

2.2 As Organizações

Passando por várias fases de como gerir uma organização (cujo ênfase se iniciou nas tarefas, seguindo nas estruturas, nos recursos humanos, no ambiente e por fim na tecnologia, desde a gestão científica até a teoria de sistemas e da contingência), hoje as organizações apresentam os seus ambientes internos como uma composição destas várias etapas.

Por sua vez, o ambiente externo de uma organização é composto pelo que o ambiente interno consegue perceber sobre os demais componentes do mundo real (natural e artificial) e dos modelos conceituais sobre este mundo real (por exemplo: conceitos de ecologia, sociedade, beleza, riqueza, juventude, mercado, etc.). A sobrevivência de uma organização depende da capacidade do ambiente interno perceber e ajustar-se ao ambiente externo através das relações (passivas e activas). O conhecimento obtido e aprendido a partir destas relações é traduzido em objectivos e acções organizacionais, possibilitando tornarem-se mais competitivas, através da inferência de novos modelos conceituais sobre o mundo real (por exemplo: novo conceito sobre riqueza, beleza, etc.) ou apenas atendendo a conceitos já existentes, ambos incorporados nos bens ou serviços disponibilizados.

2.2.1 Os componentes do ambiente interno ("*inner*") das Organizações

Com a atenção voltada para o ambiente interno ("*inner*") das organizações, em toda a actividade humana organizada vamos encontrar duas exigências fundamentais [56, 57]: a **divisão** do trabalho em várias tarefas a serem desempenhadas e a **coordenação** das mesmas. Estas duas exigências são características dos sistemas distribuídos e, para atende-las, as organizações necessitam de uma **estrutura**. A estrutura de uma organização é uma arquitectura que dispõe as unidades resultantes da divisão do trabalho, dando-lhe uma forma que assegure através da coordenação destas unidades o alcance dos objectivos. Quando estamos diante de uma actividade cujas tarefas que a compõem são executadas por uma única unidade (monolítica), e esta unidade constitui o todo para o desenvolvimento de um bem/serviço, tanto a divisão do trabalho quanto a coordenação também estão presentes, porém não de forma distribuída.

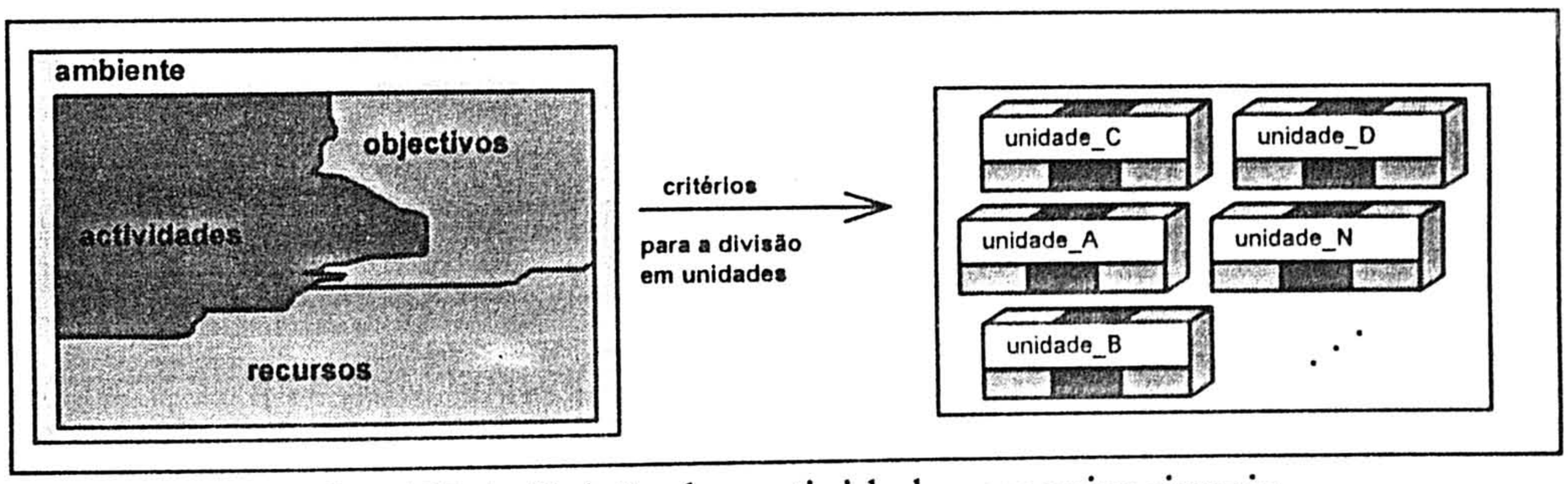
Para a **divisão** das actividades organizacionais, em unidades menores, são necessários critérios que possibilitem a partição. Esses critérios são de:

(i) **função**, devido a diferença das actividades ou recursos que utilizam,
(ii) **produto**, agrupando as diversas actividades necessárias à produção (bem/serviço) buscando intensificar os esforços e os conhecimentos necessários para o fabrico deste produto,
(iii) **cliente e mercado**, quando a satisfação das necessidades específicas de mercado, ou a melhoria do serviço prestado ao cliente se tornam objectivos importantes para a organização,
(iv) **geográfico**, utilizado por organizações que estão dispersas por extensas áreas geográficas,
(v) **projecto**, quando o produto envolve grande concentração de recursos ou as organizações são fortemente influenciadas pelo desenvolvimento tecnológico,
(vi) **processo**, quando a disposição dos recursos é a razão da agregação. É a sequência do processo produtivo que define o agrupamento, e
(vii) **matriz** é a congregação do critério função e produto para atender a uma encomenda de um cliente.

O resultado da aplicação destes critérios [fig. 18], produz unidades que devem atender: ao planeamento organizacional, à coordenação das actividades, à produção dos produtos fim (bens/serviços) e dos serviços meio. Essas unidades podem ser classificadas em duas classes:

(i) as unidades administrativas, e
(ii) as unidades executoras.

As unidades administrativas são responsáveis pelo planeamento e coordenação das actividades diferenciadas pela divisão do trabalho, pela estrutura física e conceptual. As unidades operacionais são responsáveis pela execução das actividades planeadas.



[fig. 18] A divisão das actividades organizacionais

A divisão da actividade organizacional em unidades menores, conduz a uma disposição **hierárquica** das mesmas, numa perspectiva sistémica. Duas justificações para tal disposição hierárquica são apresentadas por Simon [58]. Uma é dada pela metáfora de dois construtores de relógios os quais necessitam de uma grande quantidade de peças para montá-los. Um dos relojoeiros constrói o relógio por partes, que são resultados da divisão do todo em subsistemas. O outro opta em não dividir o todo em subsistemas e monta-o de forma monolítica. A questão da vantagem apresentada pela montagem por subsistemas é observada quando o serviço é interrompido. O relojoeiro que adopta a abordagem monolítica tem que repensar e

recomeçar muitas vezes o trabalho do ponto inicial, enquanto o da outra abordagem recomeça apenas no subsistema no qual estava a trabalhar, permanecendo os demais com o objectivo já alcançado. A outra justificação diz respeito à quantidade de informação a ser transmitida e tratada, sendo em menor quantidade nos sistemas que apresentam uma estrutura hierárquica (hierarquia do ponto de vista da divisão da actividade total em partes) do que nos demais sistemas.

Outra componente que também conduz a uma disposição hierárquica é relativa à responsabilidade pela visão do todo (maior granularidade) organizacional. Toda a tarefa antes de ser executada, necessita ser planeada, e antes de ser planeada, há necessidade da definição dos objectivos desejados e dos conceitos a serem incorporados. Para ambas as questões, surge a necessidade da existência de unidade(s) que seja(m) responsável(eis) pela geração de planos (estratégicos e tácticos) [59] e de macro-políticas. Estes planos e políticas actuam como padrões orientadores do comportamento das demais unidades, distribuindo respectivamente, as responsabilidades da coordenação e execução pelas unidades administrativas e pelas unidades executoras. Quanto à segunda questão relativa à **arquitectura**, esta diz respeito a forma como essas unidades podem ser dispostas (relacionadas) de forma que possibilitem o alcance dos objectivos através da cooperação entre cada uma delas.

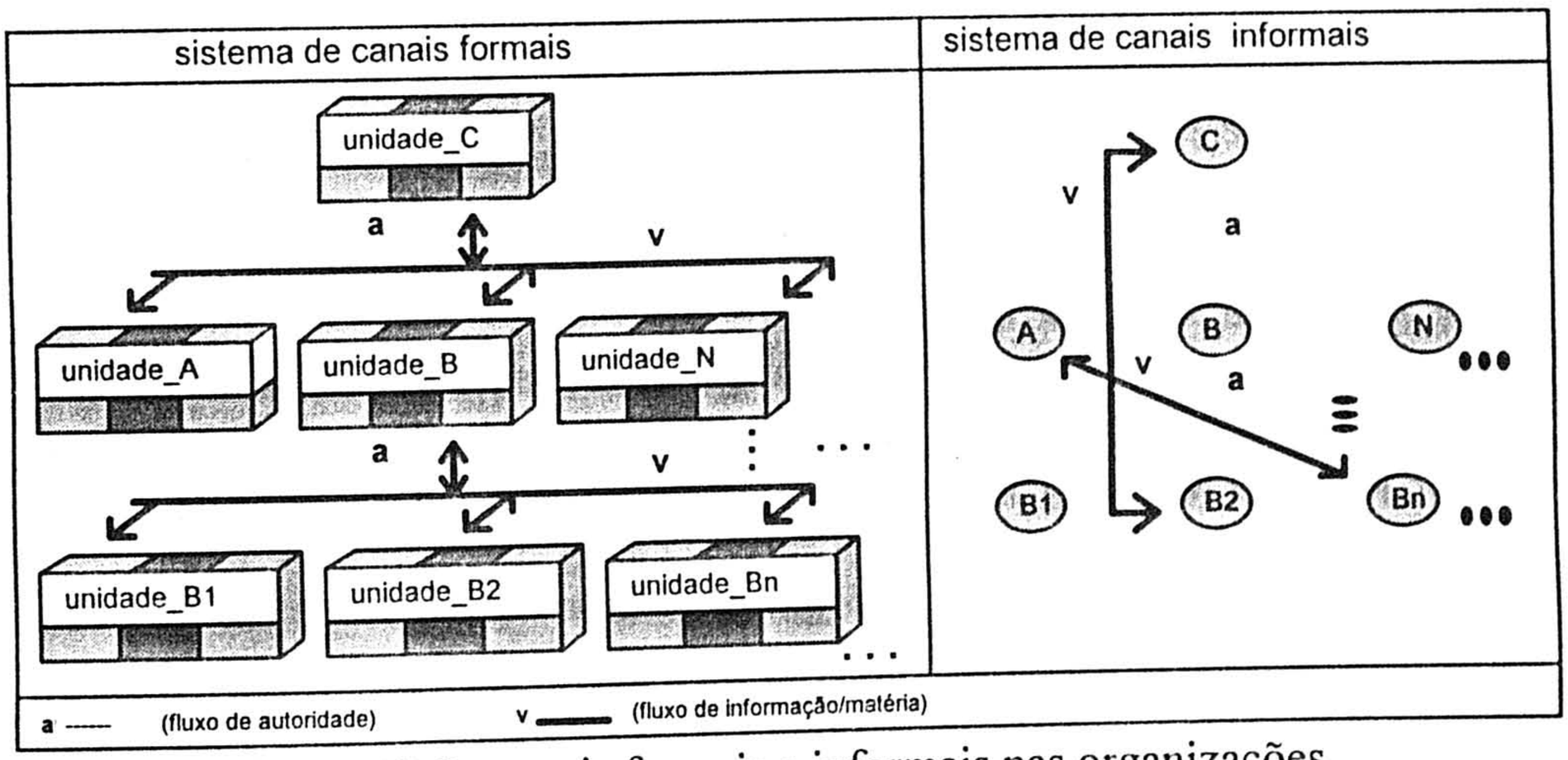
As várias unidades da organização estão relacionadas entre si por **canais**:

(i) os formais, e

(ii) os informais.

Os canais formais dizem respeito aos fluxos formalizados e regulados de autoridade (poder de tomar decisões e faze-las cumprir, interferindo em comportamentos específicos) e de informação/matéria.

Os canais informais dizem respeito aos fluxos não formalizados e não regulados, i. e. o fluxo de informação/matéria e mesmo de autoridade, gerados pelos sentimentos e interesses individuais os quais, no entanto, também se apresentam como indispensáveis ao funcionamento da parte formal da organização. Por exemplo, a [fig. 19] apresenta um conjunto de unidades, produto da divisão da totalidade das actividades organizacionais.



[fig. 19] Os canais formais e informais nas organizações

Num sistema de canais formais, o fluxo de autoridade (representado na figura pela letra "**a**") também acentua a disposição hierárquica das unidades, na qual a unidade_C orienta e decide sobre alguns dos comportamentos das unidades: A, B e N.

Por sua vez a unidade_B orienta e decide sobre alguns comportamentos das unidades: B1, B2 e Bn. No mesmo sistema de canais formais e na situação de um fluxo de informação (representado na figura pela letra "**v**"), por exemplo informar sobre o estado de uma determinada tarefa, é efectuado utilizando o canal definido para tal, em que parte coincide com o canal do fluxo de autoridade.

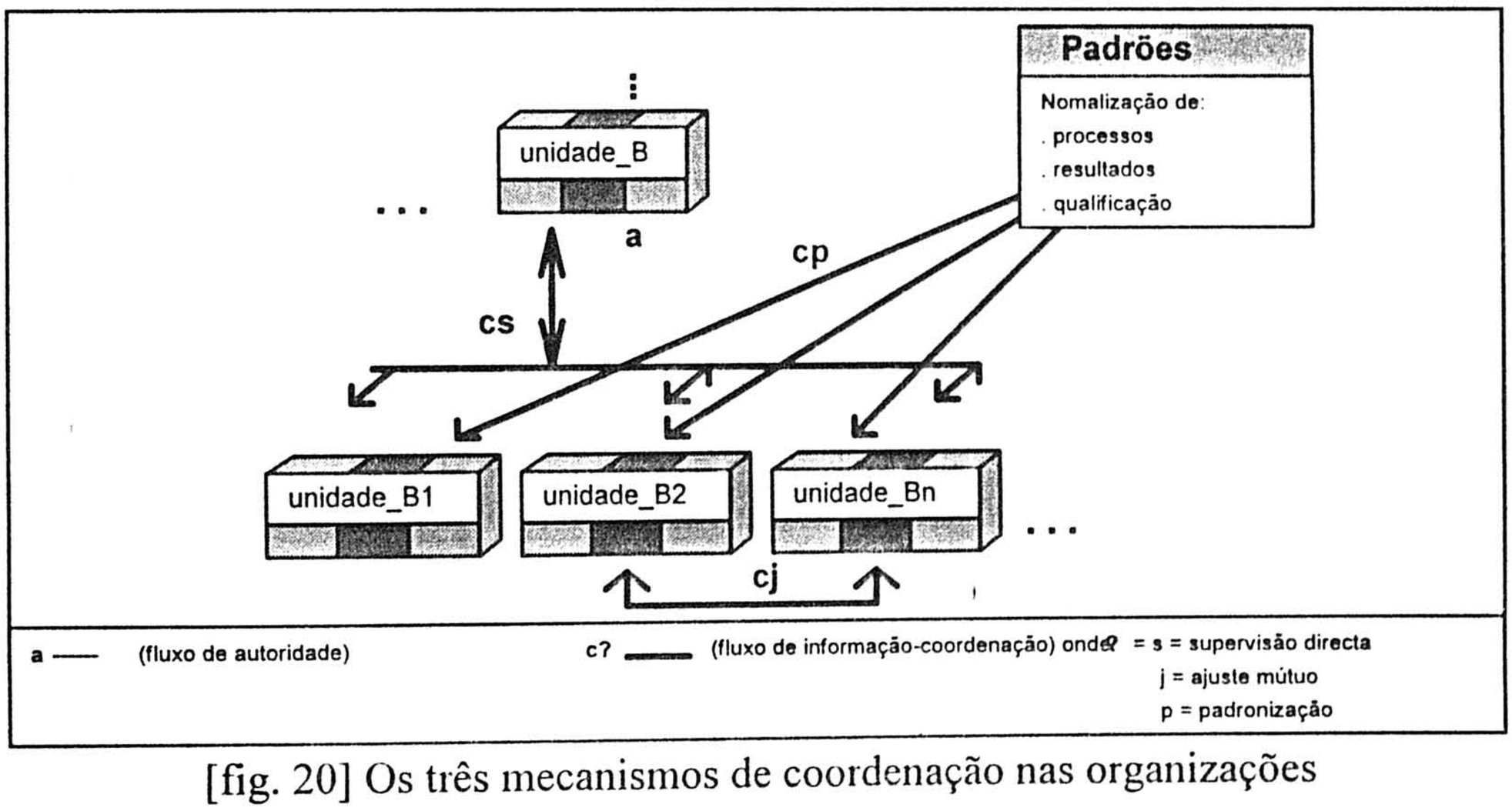
Já no caso do sistema de canais informais, há fluxo de informação que não respeita o canal formal definido para esse efeito, por exemplo a unidade_Bn comunica-se directamente com a unidade_A ou a unidade_B2 comunica-se directamente com a unidade_C.

Ainda em relação a questão da arquitectura, na parte relativa à coordenação das unidades, de forma que cada uma coopere para o alcance dos objectivos globais, mecanismos [60] são necessários para assegurar a existência desta cooperação.

A **coordenação**, para a teoria organizacional, é o processo de gerir as dependências entre actividades, a qual envolve: a tomada de decisão sobre recursos e a comunicação. A coordenação possibilita a integração e o ajustamento de esforços do trabalho individual para o cumprimento de um objectivo [61]; é a acção de trabalhar em conjunto [62].

Para a ciência da computação a coordenação está associada ao partilhamento dos recursos computacionais, à segmentação e atribuição de tarefas para os processadores e à gestão dos fluxos de informação, particularmente tratado pela Inteligência Artificial Distribuída.

Dependendo do mecanismo [fig. 20] empregue, diferentes canais (formais ou informais) são utilizados. A seguir apresentamos três de entre os mecanismos utilizados nas organizações para a coordenação das actividades internas:



[fig. 20] Os três mecanismos de coordenação nas organizações

(i) **coordenação por ajustamento mútuo**. Neste mecanismo, as unidades que se encontram relacionadas, por exemplo através de um fluxo de informação/matéria, buscam por si adaptarem-se a novas situações. Situações de conflito [63] podem surgir durante o ajustamento, sendo as metodologias de negociação ou de argumentação as utilizadas para buscar um acordo ou uma compreensão. O fluxo de informação utiliza o canal informal,

(ii) **coordenação por supervisão directa**. Neste mecanismo pressupõem-se a existência de unidades orientadoras e controladoras das actividades que são executadas pelas demais unidades com elas relacionadas. O fluxo de coordenação para este caso utiliza o canal formal, e

(iii) **coordenação por padronização**. Este tipo de coordenação ocorre pela incorporação das diversas actividades executadas pelas unidades em um programa de trabalho (descrição do processo). Neste caso, cada unidade sabe, com base no programa de trabalho, quais as suas tarefas e como as deve executar, tendo conhecimento do que exactamente esperar das demais unidades. O canal para o fluxo dessa informação é o canal formal.

Na coordenação por padronização, três tipos de padrões, baseados na normalização, são utilizados:

(i) processos de trabalho,

(ii) resultados, e

(iii) qualificações.

A normalização dos processos diz respeito a como determinada actividade deve ser desenvolvida. A normalização de resultados abrange a especificação do resultado ou desempenho a ser obtido (relativa a qualidade esperada). A normalização das qualificações diz respeito a formação dos executores de maneira que esta qualificação de "*per se*" conduza ao comportamento esperado.

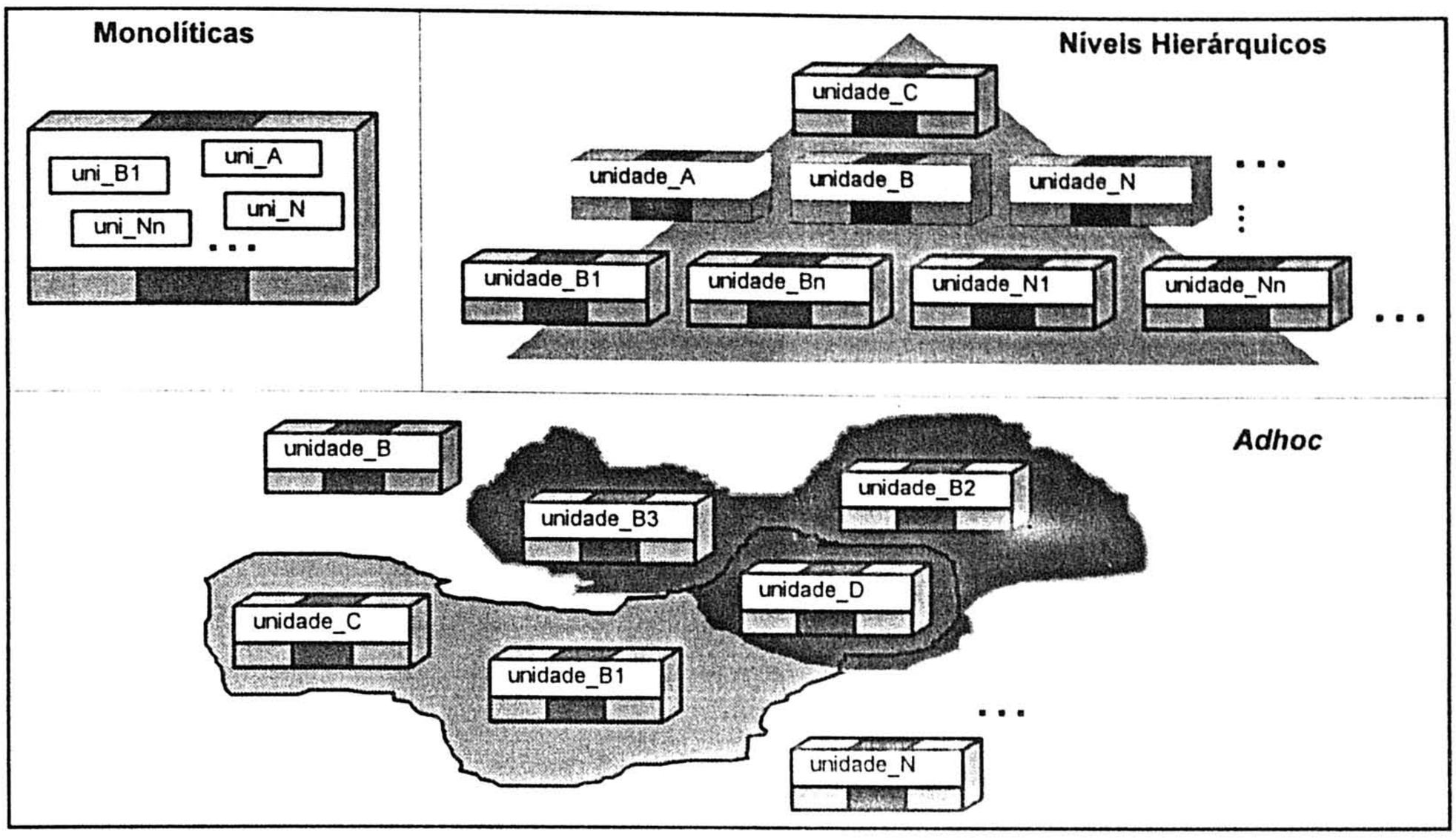
Então, o mundo interior ("*inner*") de uma organização é modelado a partir dos seguintes componentes:

(i) a **divisão** das tarefas que compõem as actividades da organização,

(ii) os mecanismos de **coordenação**, e

(iii) os canais de **comunicação**,

cujas combinações variam [fig. 21] desde uma estrutura monolítica, passando por estruturas hierárquicas com diferentes graus de centralização/descentralização até às estruturas totalmente "*adhoc*". As estruturas "*adhoc*" baseiam-se fundamentalmente no mecanismo de coordenação de ajuste mútuo e na descentralização, conduzindo a que os objectivos, a alçada de poder e da tomada de decisão sejam efectivados através da **negociação** e da **argumentação**, intensificando o fluxo de informação.



[fig. 21] As estruturas organizacionais

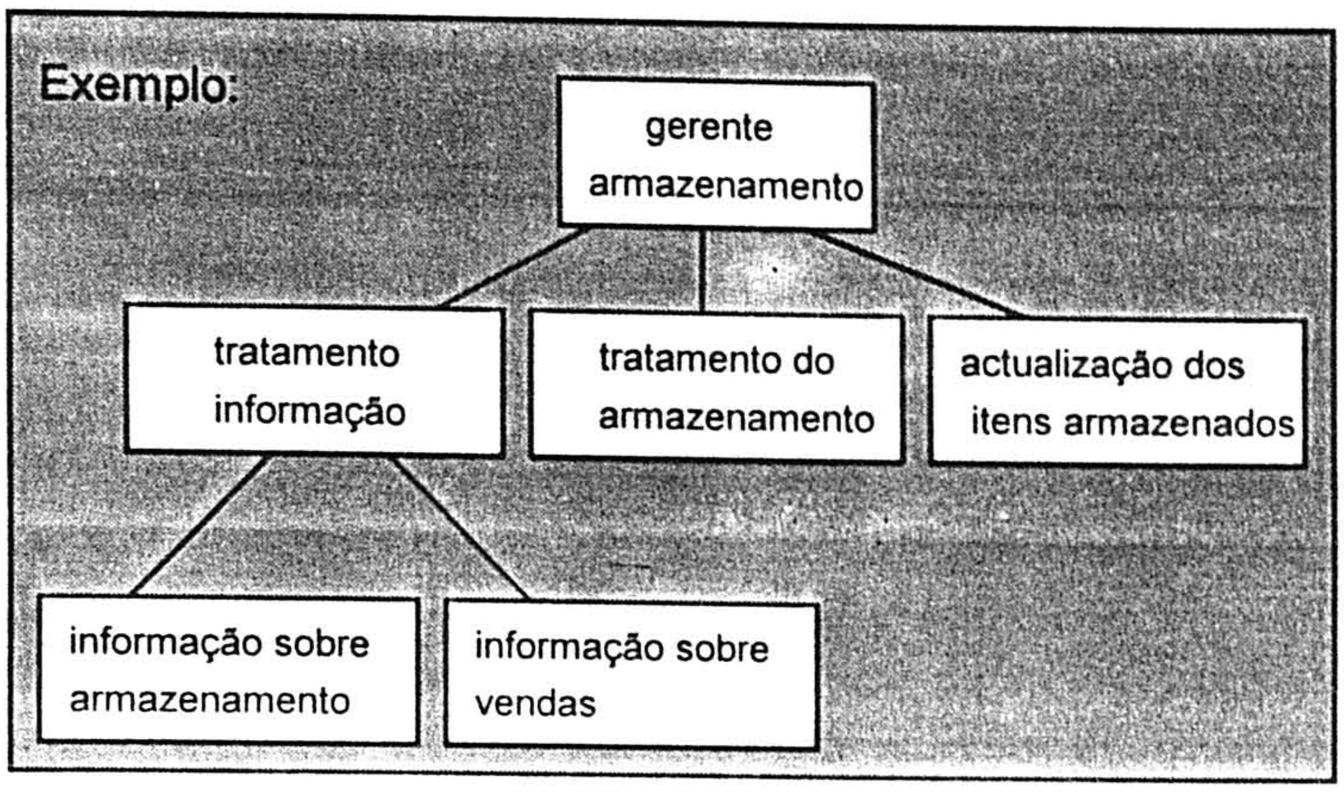
2.2.2 Arquitecturas do Ambiente Interno "*inner*" das Organizações

Um subconjunto da comunidade de investigadores no domínio da Inteligência Artificial Distribuída tem concentrado esforços na modelação das estruturas e das relações intra-organizacionais. Devidos a seus contributos para o domínio das organizações, alguns desses modelos (modelos propostos por "*Skarmeas*" e por "*Barbuceanu* e *Fox*", o modelo "*ADEPT*" ("*Advanced Decision Environment for Process Tasks*") e o modelo "*OSD*" ("*Organizational Self-Design*")) são a seguir, apresentados.

2.2.2.1 Modelo de "*Skarmeas*"

Skarmeas [64] apresenta um modelo para **automação de escritórios**, representado pela sua estrutura organizacional, o qual é baseado em sistemas multiagentes onde cada agente exibe um comportamento relativo ao papel social que desempenha ("*role theory*") na organização. De acordo com esta aproximação, um ambiente de um escritório [fig. 22] é representativo de uma organização e determinante de um conjunto de posições sociais e de uma estrutura.

As posições sociais são denominadas de papeis agregadores ("*aggregate roles*"), os quais impõem uma estrutura hierárquica, associando um conjunto de sub-responsabilidades chamadas de subpapeis ("*sub-roles*"), e por fim a cada subpapel estão associados os subpapeis elementares ("*elementary sub-roles*"). Estes subpapeis elementares descrevem unidades que contém tarefas elementares, dados e ferramentas.



[fig. 22] A estrutura de um escritório

A seguir é apresentada a descrição em "*IC-Prolog II*" [65], de alguns dos factos citados no exemplo acima [fig. 22].

```
gerente_armazenamento:{
        tipo_papel( gerente_armazenamento, agregador).
        contém( [ tratamento_informação], sub_papel)  }.
tratamento_informação:{
        tipo_papel( tratamento_informação, sub_papel).
        contém( [], sub_papel).
        contém([informação_sobre_armazenamento,
                informação_sobre_vendas], papel_elementar) }.
pessoa( jim):{
        contém([gerente_armazenamento], agregador) }.
hierarquia:{
        superior(gerente_empresa, gerente_compras).
        superior(gerente_compras, gerente_armazenamento).
        superior(gerente_armazenamento, gerente_vendas) }.
```

Para este sistema, as tarefas são classificadas e declaradas como tarefas simples, tarefas complexas e tarefas remotas.

Uma tarefa simples é uma tarefa em sua unidade atómica (não decomponível) e passível de realização. Uma tarefa complexa é uma tarefa que necessita de decomposição, tornando-se um conjunto de tarefas simples. O conhecimento sobre a decomposição está presente na própria descrição da tarefa complexa. Uma tarefa remota é uma tarefa que não pode ser decomposta ou executada localmente, porém pode ser executada por outro agente. Devido a este sistema apresentar a estrutura organizacional de forma explícita, ele pode ser facilmente ajustável as mudanças estruturais organizacionais.

A arquitectura multiagente consiste de três partes: uma concha ("*shell*"), a base de conhecimento e o executor. A concha é responsável pelo tratamento das mensagens recebidas e pela execução da tarefa. Ela contém quatro componentes que se comunicam entre si:

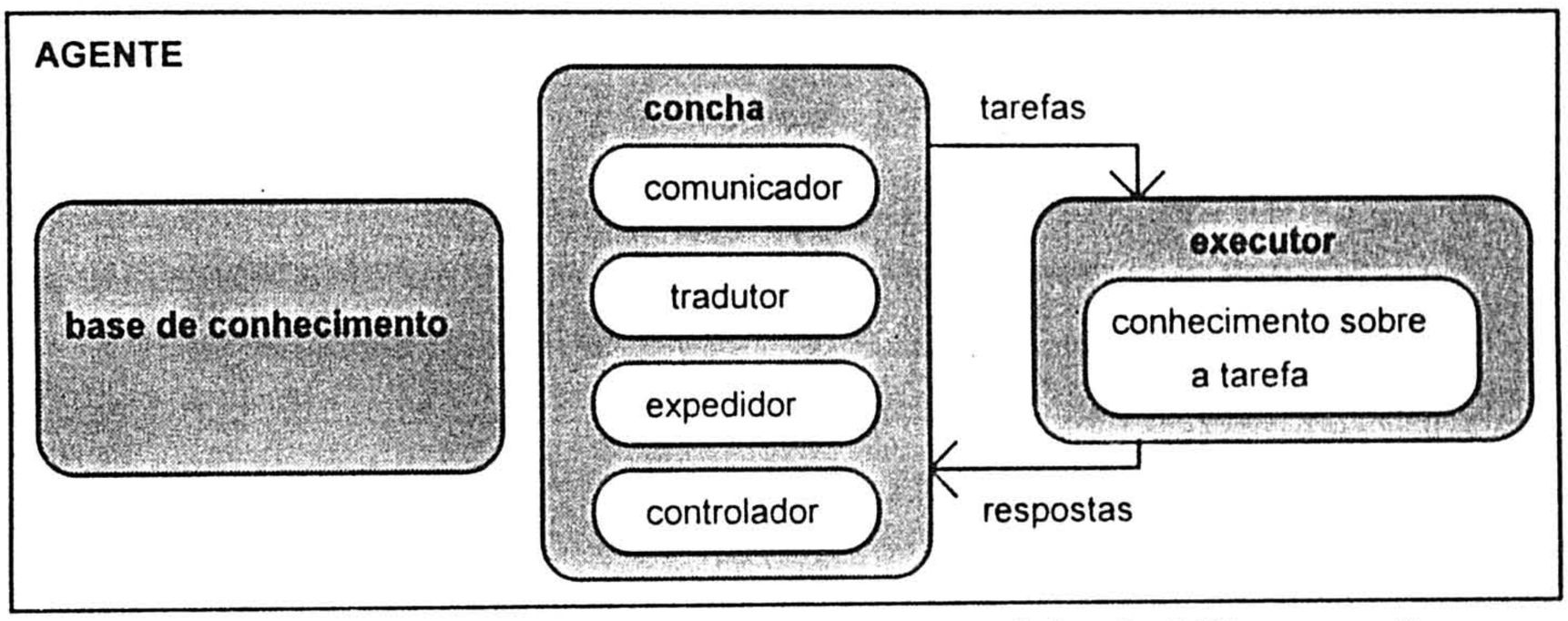
(i) o comunicador, o qual é responsável pela comunicação com o exterior,

(ii) o tradutor de mensagens, o qual é responsável pela conversão das mensagens recebidas. Esta conversão significa transformar a mensagem recebida em objectivos a serem alcançados,
(iii) um expedidor, o qual é responsável pelo escalonamento das tarefas de acordo com o significado da mensagem recebida, e
(iv) um controlador, o qual controla a execução da tarefa.

Um segundo módulo, a base de conhecimento, contém informações sobre os papeis e sobre as tarefas. O terceira módulo, o executor, contém o conhecimento sobre a tarefa a ser executada.

Uma das principais características deste tipo de aproximação, é o facto de ser de aplicação independente do domínio, sendo a base de conhecimento a única componente a ser alterada.

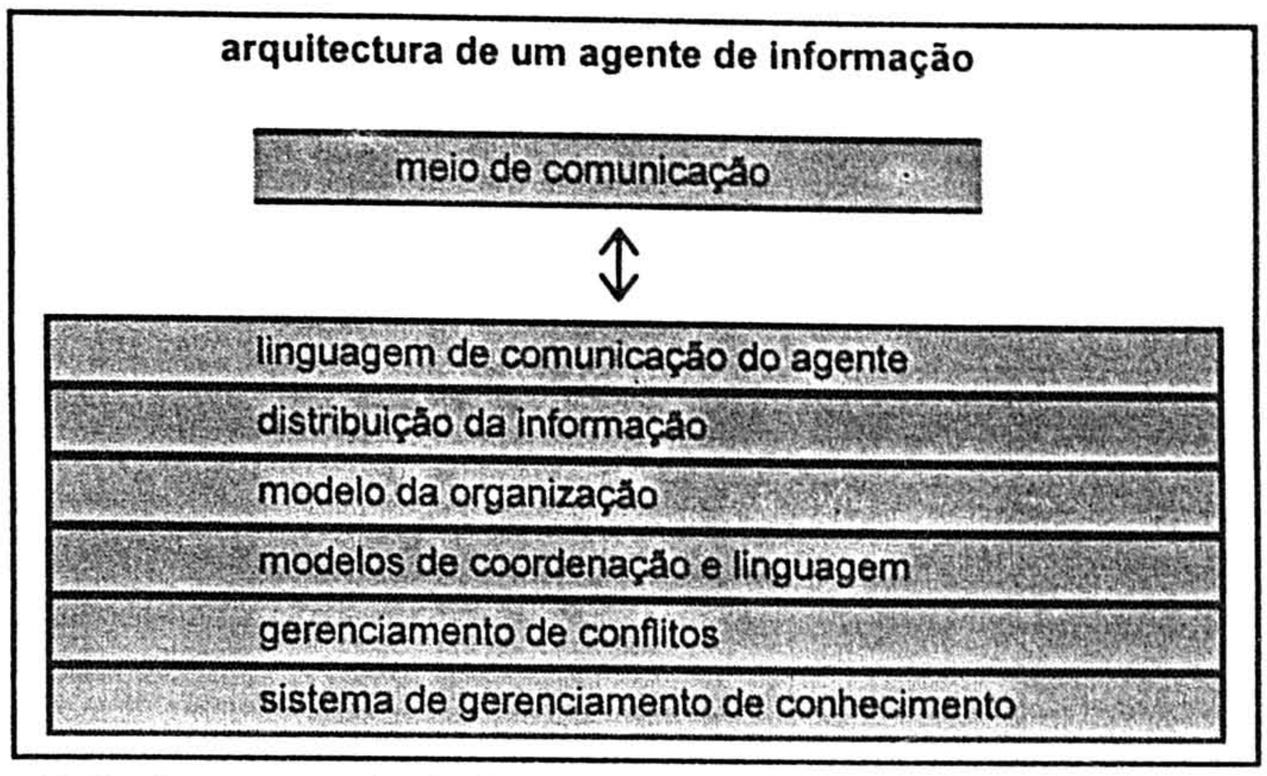
Quanto à comunicação entre diferentes agentes, esta é feita através de "caixas de correio" ("*mailboxes*"). Quando um agente é inicializado, uma caixa de correio é criada e a ele associada. A figura a seguir [fig. 23] apresenta a arquitectura acima descrita.



[fig. 23] A arquitectura do agente no modelo de "*Skarmeas*"

2.2.2.2 Modelo de "*Barbuceanu* e *Fox*"

Barbuceanu e Fox [66] apresentam uma arquitectura para a actividade cooperativa em empresas, integradas através de uma classe de agentes denominada de agentes de informação ("*information agents*"). Estes agentes são enquadrados como uma das componentes da infra-estrutura de informação em ambientes computacionais baseados em conhecimento cooperativos. Para alcançar um estado de respostas rápidas face às mudanças do mercado, através da restruturação dos seus negócios, as organizações encontram nos sistemas de informação cooperativos, auxílio para providenciar o acesso a informação, para a tomada de decisão e para a execução de acções. Os agentes de informação assumem uma substancial quantidade do trabalho requerido para executar a comunicação com outros agentes, para procurar, converter e melhorar a informação e para coordenar o processo de distribuição dessa informação. Para alcançar esses objectivos os agentes de informação apresentam os seguintes componentes [fig. 24]:

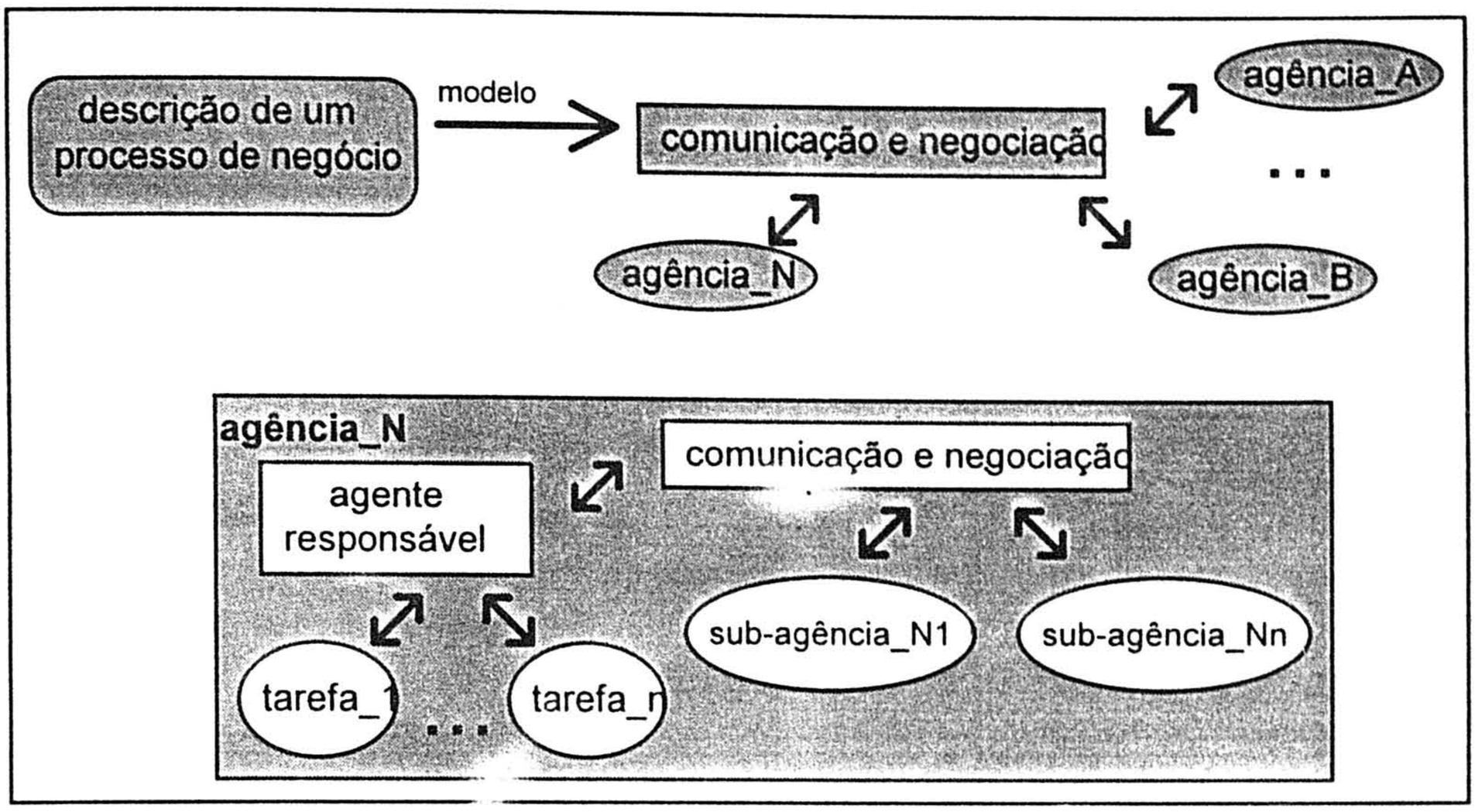


[fig. 24] O agente de informação no modelo de "*Barbuceanu* e *Fox*"

(i) **linguagem de comunicação do agente**. É a linguagem de comunicação dos agentes, cujos actos de discurso ("*speech acts*") utilizam a linguagem "*KQML/KIF*" [117, 118],
(ii) **distribuição da informação** é o serviço que distribui de forma voluntária ou a pedido, informações de interesse para os outros agentes,
(iii) **modelo da organização** é o modelo contido em cada agente, o qual descreve os papeis que os demais agentes desempenham na organização, os seus objectivos e capacidades,
(iv) **modelos de coordenação e linguagem**. Esses modelos contém regras declaradas numa linguagem para coordenação das acções cooperativas,
(v) **gerência de conflitos** capacita o agente a tomar decisões quando confrontado com informação contraditória recebida de outros agentes, e
(vi) **sistema de gestão de conhecimento**, o qual armazena e processa o conhecimento localmente. Este processamento, por exemplo, baseia-se em inferência dedutiva, revisão de crenças, etc.

2.2.2.3 Modelo "*ADEPT*"

Outro sistema que apresenta também uma arquitectura multiagente para a gestão de processos de negócios em organizações é o do projecto "*ADEPT*" ("*Advanced Decision Environment for Process Tasks*") [67, 68]. O sistema "*ADEPT*" consiste num número de **agências** autónomas, possibilitando modelar a estrutura organizacional como hierárquica, plana ou uma combinação de ambas. O modelo de cada sistema envolve a transformação da descrição dos processos de negócios num número de agências, cada uma com conexão através de um meio de comunicação comum, i.e. uma rede de comunicação de extensão local, regional ou mundial. Uma **agência** consiste num agente responsável, num conjunto de tarefas que este agente responsável pode executar e num conjunto de sub-agências [fig. 25].



[fig. 25] As agências do modelo "*ADEPT*"

O agente responsável de uma agência representa os interesses da agência em relação a outras agências, sendo qualquer comunicação interagência executada através deste. Uma subagência tipicamente apresenta um comportamento cooperativo relativo ao seu agente responsável. A relação entre subagências e agente responsável pode ser vista como um tipo de compromisso social, o qual não pode ser recusado.

Uma subagência é uma agência subsidiária com um alto grau de autonomia local, porém necessariamente coopera com o agente responsável.

Para um agente participar no ambiente "*ADEPT*" é necessário que o mesmo utilize uma expressiva **linguagem** comum de comunicação. A justificação para o uso de uma linguagem comum encontra-se no facto de, para uma comunidade de *n* agentes, se uma linguagem não comum for utilizada, $O(n^2)$ mecanismos de transformação entre representações de conhecimento são requeridos. Caso contrário são necessários apenas $O(2n)$. Esta linguagem comum, consiste num protocolo i.e. um conjunto de actos de discurso ("*speech acts*") e uma sintaxe para exprimir a informação.

A utilização de uma linguagem comum de conhecimento e a interpretação semântica comum de símbolos, capacitam os agentes da comunidade para:

(i) conhecer a intenção da comunicação via um conjunto de actos de discurso (informar, rejeitar ou requerer alguma proposição),

(ii) interpretar o conteúdo da mensagem através da utilização de uma sintaxe comum ("*KIF - Knowledge Interchange Format*"), e

(iii) compreender qual o significado do conteúdo da mensagem através de compromissos baseados em ontologias (qual o significado da proposição que o agente está informando, rejeitando ou requerendo).

2.2.2.4 Modelo "*OSD*"

So e Durfee [69] apresentam um modelo de auto-configuração organizacional ("*Organizational Self-Design - OSD*") para reestruturação organizacional, cujo objectivo é encontrar os factores que afectam a execução colectiva de um conjunto de agentes autónomos orientados para a resolução de um problema de forma cooperativa.

Agentes cooperantes tipicamente trabalham em ambientes sujeitos a mudanças, tornando-se importante que os mecanismos de coordenação empregues sejam capazes de se adaptarem a essas mudanças.

Na disciplina de Inteligência Artificial Distribuída, a organização adaptativa, quando executada pelos membros da organização, é denominada de auto-configuração organizacional ("*Organizational Self-Design - OSD*"). Uma organização com capacidade para "*OSD*" deve possuir um ou mais de seus membros dotados de capacidades para executarem as seguintes tarefas:

(i) **monitorização**. Tarefa de monitorar quanto a eficácia da estrutura organizacional, baseado em parâmetros que afectam o seu desempenho, bem como as condições sob as quais uma reorganização deve ser reconsiderada,

(ii) **configuração**. Identifica a nova estrutura organizacional apropriada para a nova situação. Para a geração de estruturas organizacionais, algumas importantes dimensões devem ser consideradas, tais como: a decomposição de tarefas em subtarefas, a atribuição das subtarefas para os agentes disponíveis, a determinação de estruturas de papeis e de comunicação entre agentes, a quantidade de agentes envolvidos, e quais os recursos a serem usados e como eles são partilhados entre os agentes,

(iii) **validação e selecção**. Tarefa de avaliar cada alternativa da nova estrutura, através de uma medida de desempenho e seleccionar aquela que apresentar o melhor valor, e

(iv) **implementação**. Implementar a nova estrutura seleccionada, o que requer que as tarefas sejam transferidas para outros agentes, que não os que anteriormente as executavam.

As tarefas para um "*OSD*" podem ser executadas com uma abordagem global, do geral para o específico [70] ou de uma maneira local, do específico para o geral [71]. Na primeira abordagem um agente monitora o desempenho global e as actividades locais dos membros, projecta as alternativas para uma reorganização, selecciona a melhor e impõem a reestruturação para os outros agentes. Na segunda, o "*OSD*" ocorre a nível do agente local, utilizando o que este agente conhece localmente sobre o desempenho da organização em sua vizinhança, projecta alternativas para si, selecciona uma alternativa e a adopta. Esta alteração local pode afectar o desempenho da vizinhança, causando um efeito tipo cascata, o que leva a emergir uma nova organização a partir de uma escolha local.

Neste método, a estrutura organizacional pode ser vista como uma especificação de métodos de controlo de pesquisa distribuída. A coordenação é então caracterizada como uma pesquisa através de um espaço de comportamentos [72]. Se um agente é responsável por executar o controlo da pesquisa sobre o espaço de comportamentos de todos os agentes, isto torna-se uma coordenação centralizada onde o agente de controlo procura encontrar e escolher comportamentos aos demais agentes. Por outro lado, quando múltiplos agentes participam, torna-se num problema de coordenação distribuída, o que requer a distribuição da pesquisa do espaço de comportamentos pelos múltiplos agentes.

Nesta perspectiva a "*OSD*" é caracterizada como uma procura através do espaço de comportamentos para uma provável ou melhor escolha de comportamentos entre os agentes.

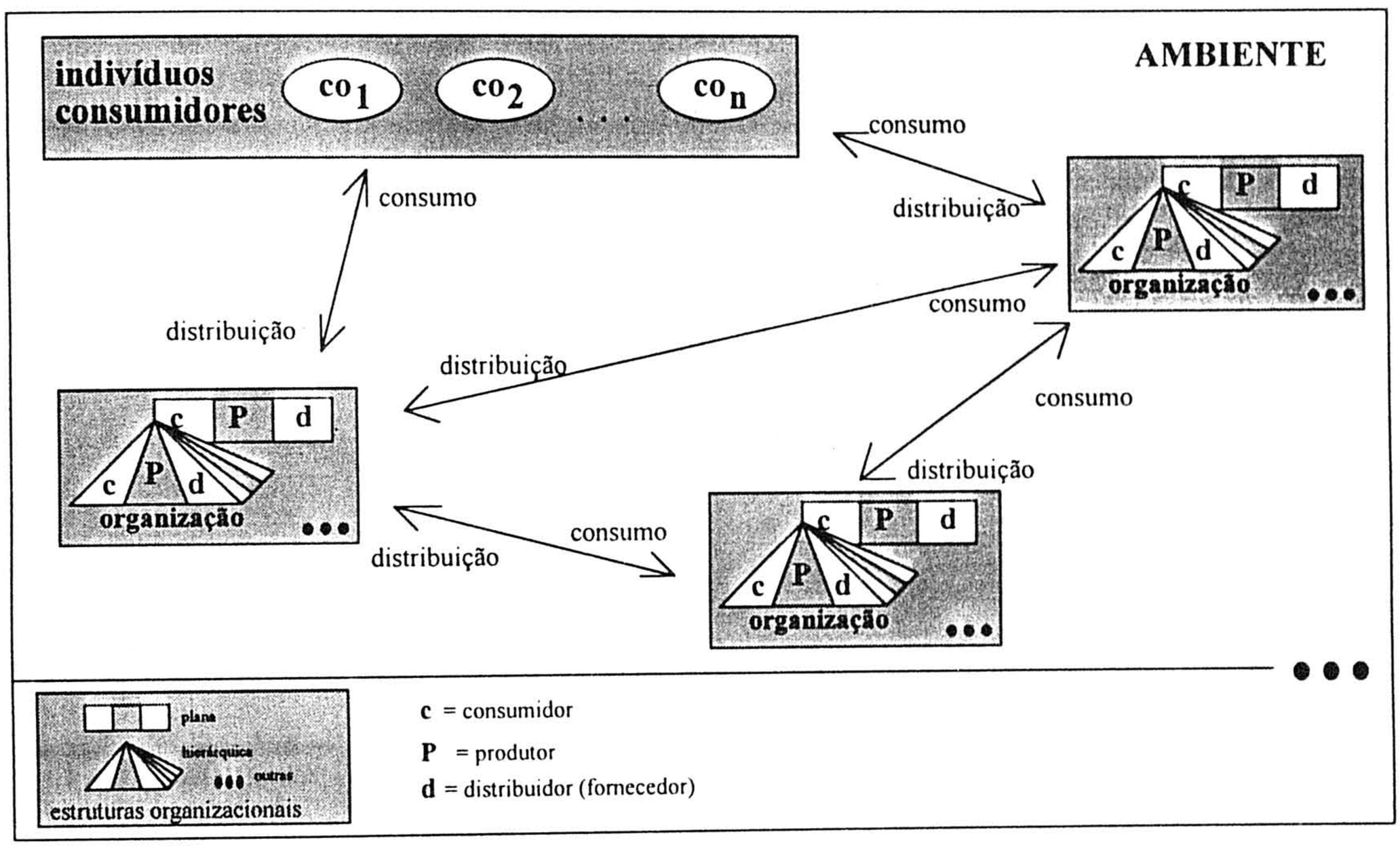
2.3 Um Cenário para uma Sociedade de Organizações

2.3.1 Tipos de "Ambientes"

O ambiente exterior ("*outer*") às organizações, como é o caso de um mercado de bens e serviços, é o ambiente composto por dois tipos de agentes económicos [fig. 26]:

(i) as organizações, e
(ii) os indivíduos,

associados através de relações sociais (são relações sociais no sentido em que os agentes raciocinam considerando o conhecimento dos demais agentes envolvidos na relação).



[fig. 26] Uma sociedade de organizações

Esses agentes, dinamicamente, assumem os papeis básicos de produtores, distribuidores e consumidores.

As relações desempenham uma parte decisiva no comportamento das organizações e dos indivíduos, e são utilizadas para:

(i) decidir o que e como produzir,
(ii) indicar onde e como distribuir a produção,
(iii) determinar como promover e validar os produtos produzidos ou os serviços prestados, e
(iv) o que consumir.

Os agentes económicos neste mercado caracterizam-se por:

(i) serem autónomos (agem de acordo com seus conhecimentos e em favor de seus interesses, tomando suas próprias decisões segundo suas crenças),

(ii) possuírem uma limitada competência e conhecimento sobre as decisões dos outros, e

(iii) por necessitarem de cooperação para o alcance de objectivos.

Nas interacções cooperativas através das quais as organizações buscam parceiros (a nível de fornecimento e distribuição, a nível de terceirização de serviços - "*outsourcing*" e a nível de criação de uma estrutura virtual), as actividades que envolvem estes processos cooperativos necessitam de mecanismos de coordenação.

Por exemplo:

(i) quando uma organização decide não fabricar determinado componente, o qual é parte de um certo produto, e opta por ir buscá-lo no mercado, adquirindo-o de uma outra organização, ou

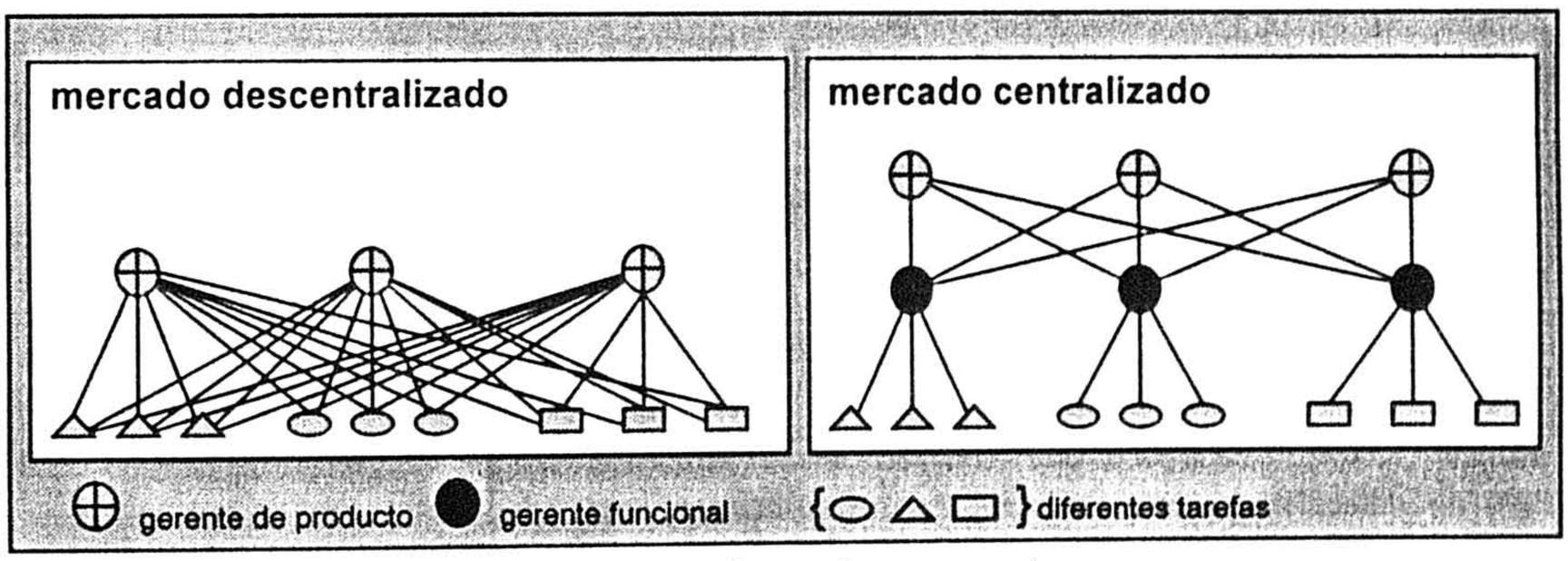
(ii) quando uma organização, decide manter apenas uma estrutura mínima interna e desenvolver seus produtos (projecto, produção e distribuição) fora desta estrutura.

Em ambos os casos a coordenação apresenta-se como um mecanismo de mercado. Neste mecanismo de mercado, encontram-se diferentes e independentes tomadores de decisão (organização interessada em contratar e organizações interessadas em serem as contratadas), interagindo uns com os outros para alcançarem um objectivo comum, sem a existência de qualquer controlador que centralize a coordenação e consequentemente as decisões.

Na definição da estrutura de coordenação dois tipos de mercados [fig. 27] são considerados [60]:

(i) o mercado descentralizado, e

(ii) o mercado centralizado.



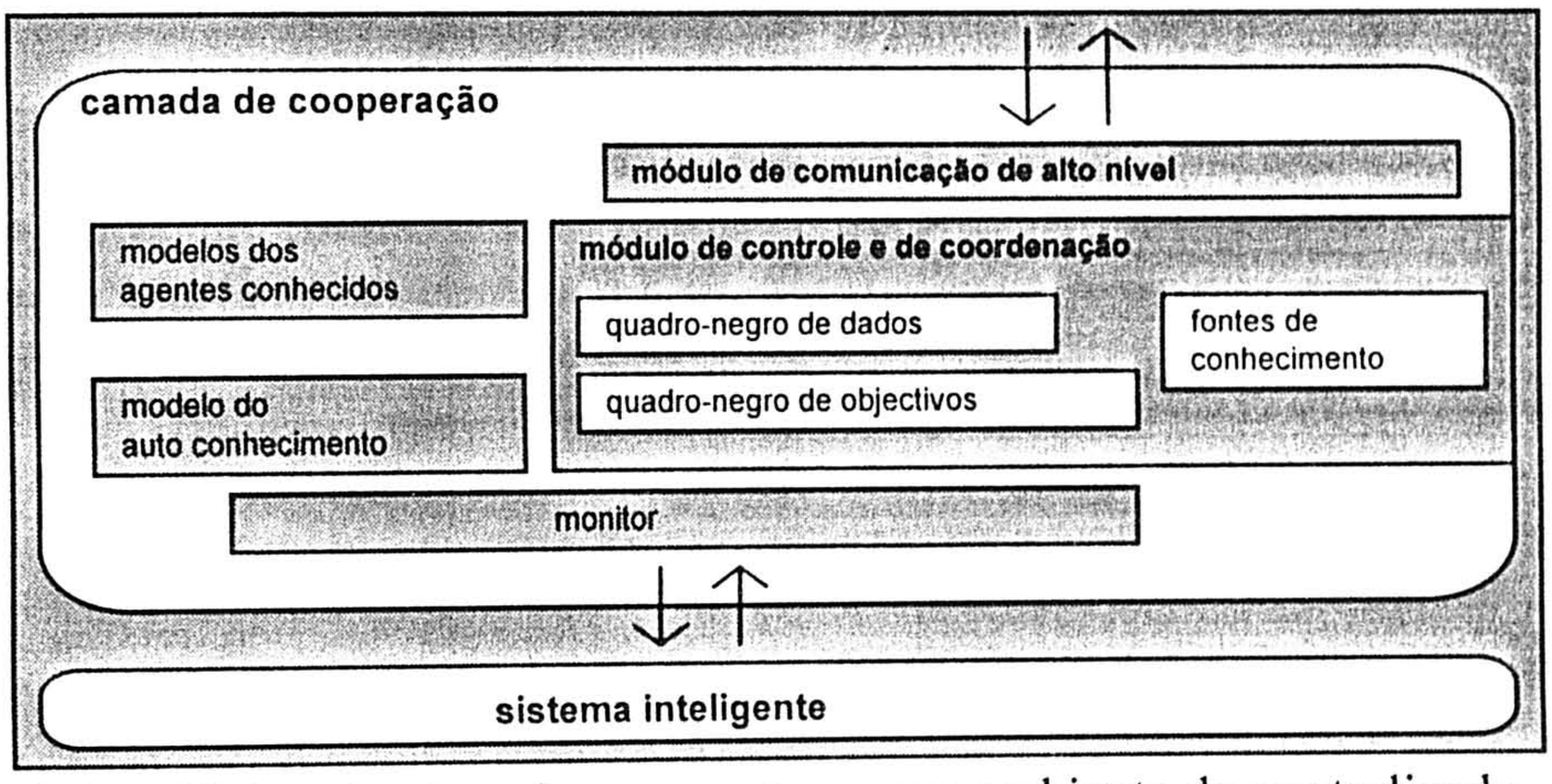
[fig. 27] Tipos de mercado

2.3.2 Mercado descentralizado

No mercado descentralizado, de forma pura, todos os compradores estão em contacto com todos os vendedores (fornecedores) e cada um deles toma suas próprias decisões sobre qual transacção aceitar ou não. Os compradores desempenham o papel de gerente de produto e enviam mensagens solicitando propostas para todos os vendedores do bem/serviço desejado. Recebem as propostas e seleccionam o fornecedor que contiver a melhor oferta (baseado em critérios de avaliação). Se existirem "*m*" fornecedores, "*2m*" mensagens serão necessárias por produto/serviço desejado. Duas mensagens adicionais são também utilizadas, uma para atribuir o pedido do bem/serviço e outra para notificar o comprador que o bem/serviço foi entregue ou está encerrado.

2.3.3 Exemplo de mercado descentralizado

Oliveira e Ramos [73, 74, 119] apresentam uma arquitectura de agente [fig. 28], para sistemas multiagentes em ambientes descentralizados, a qual segue de perto a proposta do modelo desenvolvido no projecto "*ARCHON*" [120].



[fig. 28] Arquitectura de um agente em um ambiente descentralizado

São dois os subsistemas fundamentais desta arquitectura; a camada de cooperação, a qual é responsável pela coordenação das actividades intra e interagentes, e um sistema baseado em conhecimento com capacidade para resolver problemas num domínio específico.

A camada de cooperação é composta por:

(i) **um módulo de comunicação** de alto nível para trocar informações com outros agentes,

(ii) **um módulo controlador e coordenador**. Este módulo controla as actividades intra-agente, coordena os seus próprios objectivos com os objectivos dos outros agentes, escolhe a política de cooperação e executa o protocolo correspondente, resolve conflitos e controla a comunicação,

(iii) **um monitor**, para escalonar as actividades do sistema inteligente,

(iv) **um modelo do auto conhecimento**, o qual representa as capacidades e as necessidades do agente, e

(v) **modelos dos agentes conhecidos**, os quais explicitamente representam o conhecimento (quais os agente que podem executar uma tarefa específica, quais os agentes interessados em determinadas informações e que informações o agente já forneceu) sobre os demais agentes da comunidade.

Numa sociedade composta por agentes com esta arquitectura, duas formas fundamentais de cooperação podem ser identificadas:

(i) a partilha de tarefas ("*task-sharing*"), e

(ii) a partilha de resultados ("*result-sharing*").

Estas duas formas de cooperação, são orientadoras do mecanismo de coordenação utilizado pela sociedade, caracterizando um mercado descentralizado.

Duas situações de conflito, quando da utilização do protocolo de cooperação do tipo partilha de tarefas ("*task-sharing*"), podem ocorrer:

(i) quando mais de um agente é capaz de executar a actividade proposta. Para este caso, a metodologia de negociação é utilizada, e

(ii) quando não existem agentes habilitados para a actividade. O tratamento dispensado ocorre através do replaneamento da actividade.

2.3.4 Modelos baseados na descentralização

A partir do protocolo para processos de negociação denominado de rede de contratos ("*Contract Net Protocol*") [76], dois modelos baseados na descentralização foram desenvolvidos por Kuhn e Müller [77]:

(i) modelo de decomposição descentralizada de tarefas ("*Decentralized Task Decomposition Model*") e

(ii) modelo totalmente descentralizado ("*Completely Decentralized Model*").

No protocolo rede de contractos, a atribuição de subtarefas é efectuada por meio de um agente decisor o qual além de decompor a tarefa em subtarefas, anuncia-as para a sociedade de agentes e promove a escolha do agente a ser contratado a partir das ofertas recebidas. Durante este processo o agente decisor, assume o papel de gerente e os demais agentes que a princípio aceitaram (podiam ter recusado) a proposta, assumem papéis de subordinados.

No modelo de decomposição descentralizada de tarefas, é a sociedade de agentes que deverá decompor a actividade em tarefas, atribuir as tarefas e elaborar um plano para a execução da actividade, sem a ajuda de um agente desempenhando o papel de gerente. Os agentes podem propor um plano total ou parcial para os demais agentes ou podem elaborar um plano em conjunto. Este plano pode ser obtido através de fazer circular uma mensagem de agente para agente até o plano completo estar produzido, o qual é aceite por todos os agentes participantes. Porém existindo conflitos entre os potenciais contratados, a negociação pode ser considerada como solução para esses problemas. Os potenciais contratados podem negociar entre si. Esta solução híbrida mantém o papel do agente gerente para anunciar a actividade, receber as tarefas, e sintetiza-las, deixando o processo de negociação para os demais agentes interessados em ser os contratados.

Para o modelo totalmente descentralizado, o agente que deveria desempenhar o papel de gerente torna-se desnecessário. A sociedade de agentes é que decompõem a actividade num conjunto de subtarefas e os agentes combinam a solução para o plano

global dessa actividade. Este processo também é desenvolvido através da negociação, onde cada agente mantém o modelo de conhecimento dos demais agentes, sendo este conhecimento actualizado a partir das mensagens recebidas e através da percepção do comportamento dos mesmos. Os agentes podem criar planos para as actividades ou para as subtarefas e enviá-los para os demais, os quais podem aceitá-los, rejeitá-los, refiná-los ou modificá-los e finalmente concordarem num plano conjunto. Estas duas extensões do protocolo de rede de contratos foram aplicados no cenário *MARS* ("*Modeling a Multi-agent Scenario for Shipping Companies*") [78] para companhias de transportes.

2.3.5 Mercado centralizado

No mercado centralizado, os compradores não necessitam estar em contacto com todos os possíveis vendedores, devido à existência de um intermediário o qual mantém este contacto. Para esta situação, menos conexões e mensagens são necessárias em relação ao mercado descentralizado. O intermediário desempenha o papel de gerente funcional, o qual coordena todas as transacções de compra e de venda.

Quatro mensagens, para a coordenação, são requeridas: uma originária do comprador para notificar o intermediário da sua necessidade de cooperação, uma por parte do intermediário para direccionar o pedido ao vendedor e duas para notificar o intermediário e o comprador respectivamente que a tarefa está completa.

2.3.6 Modelo "*WALRAS*"

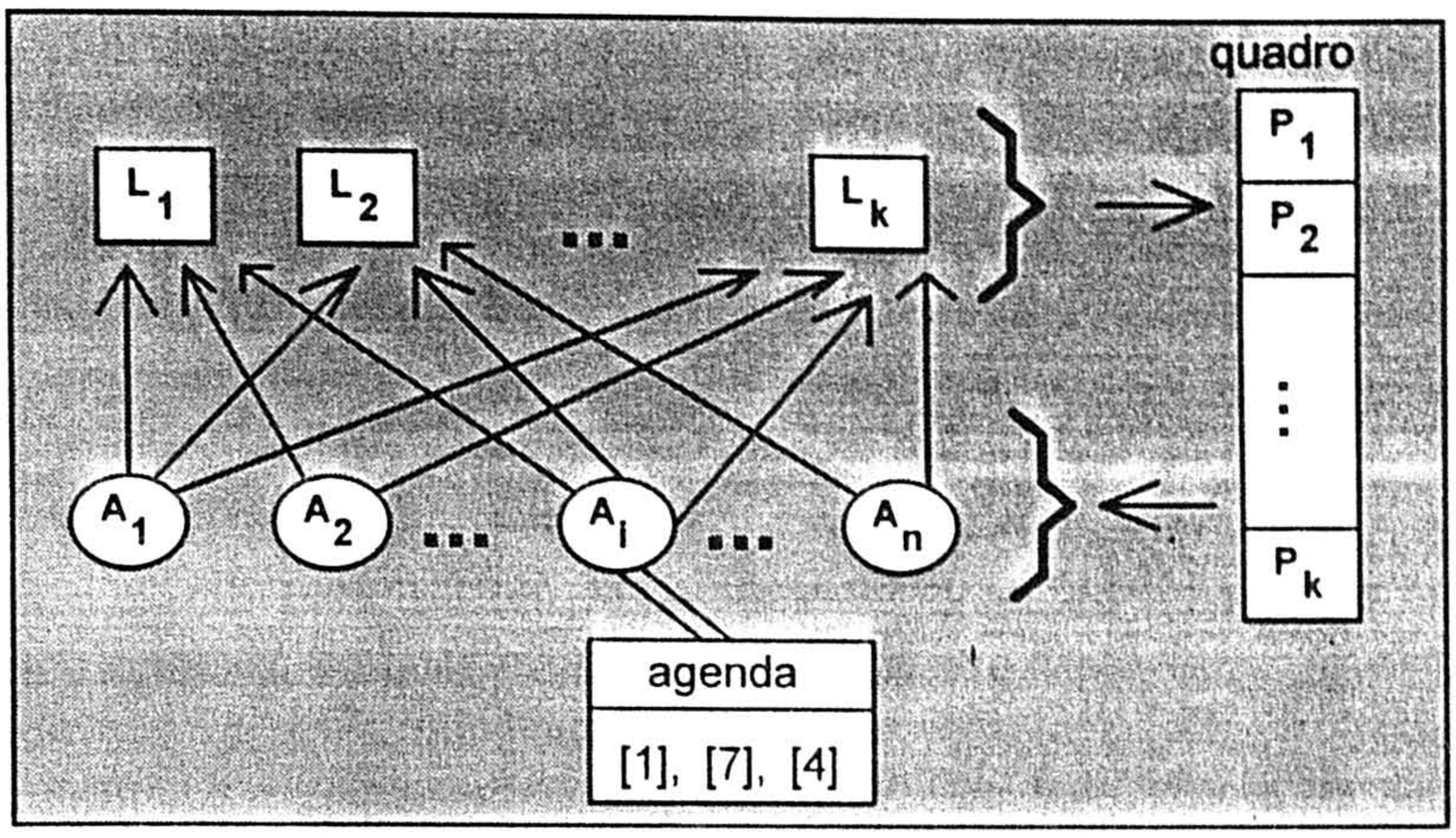
Nos mercados, os sistemas de preços constituem uma classe de mecanismos os quais providenciam uma efectiva descentralização de decisões com o mínimo de comunicação. O sistema "*WALRAS*" apresentado por Wellman [75] representa uma aproximação geral para a construção e análise de sistemas de planeamento distribuído, baseados na teoria geral do equilíbrio e mecanismos de competitividade. Este sistema trabalha pela obtenção de um equilíbrio correspondente a uma particular configuração de agentes e de mercadorias. O método utilizado no "*WALRAS*" baseia-se no descrito por Léon Walras (economista Francês 1834-1910) para explicar como os preços podem ser obtidos. Partindo de um vector de preços, o método consiste em:

(i) cada agente determina suas encomendas para os preços presentes no vector,

(ii) os agentes informam a quantidade encomendada para um leiloeiro, e

(iii) baseado nas informações recebidas dos agentes sobre as encomendas para os preços inicialmente contidos no vector, o leiloeiro iterativamente ajusta os preços para cima ou para baixo caso esteja em excesso a encomenda ou o fornecimento, respectivamente.

A [fig. 29] apresenta o processo de oferta implementado no sistema "*WALRAS*".



[fig. 29] O processo de ofertas no sistema "*WALRAS*"

Existe um leilão ($\mathbf{L_x}$) distinto para cada mercadoria. Para cada agente existe uma ligação para todos os leilões para os quais os agentes estão interessados. Existe um quadro para os preços correntes, o qual é mantido actualizado pelos vários leilões e cada alteração é enviada a todos os agentes interessados, podendo então esses valores serem dispensados. Cada agente mantém uma agenda, especificando em quais mercados ele deve actualizar suas ofertas ou calcular uma nova. Na figura acima, o agente $\mathbf{A_i}$ tem pendente tarefas para submeter ofertas para os leilões $\mathbf{L_1}$, $\mathbf{L_7}$ e $\mathbf{L_4}$. O processo de ofertas é altamente distribuído e cada agente necessita apenas comunicar directamente com o leilão das mercadorias de seu interesse. Não há coordenação entre leilões e os agentes não necessitam negociar e nem se conhecer uns aos outros.

As novas ofertas são recebidas no leilão e os preços previamente obtidos tornam-se obsoletos. Periodicamente, cada leilão efectua o cálculo do novo preço (se novas ou actualizadas ofertas são recebidas) e coloca-o no quadro. Uma vez que este preço afecte a oferta do agente, este incrementa a sua agenda com a identificação do leilão correspondente. Enquanto os agentes não tiverem as suas agendas vazias ou houver leilões com novas ofertas, algumas ou todas as mercadorias podem estar em **desequilíbrio**, caso contrário, a economia está em **equilíbrio competitivo** (com alguma tolerância numérica).

2.3.7 Conclusão

Uma característica fundamental que difere o interior ("*inner*") de uma organização do seu exterior ("*outer*"), encontra-se na questão relativa à coordenação. No interior de uma organização os agentes que dela participam possuem diferentes graus de autonomia na execução de suas actividades, na negociação ou na argumentação. As solicitações não podem ser recusadas sem uma legítima razão, pois os objectivos globais da organização não podem ser comprometidos. Por outro lado, no exterior de uma organização, cada agente da sociedade (organização ou indivíduo) é autónomo nas suas decisões (salvaguardando as regras, tais como as advindas da autoridade constituída, por exemplo o Estado), agindo em seu próprio benefício, motivado pelos seus interesses.

Os ambientes exteriores onde os agentes de uma sociedade multiagente interagem podem ser classificados com descentralizados e centralizados. O critério fundamental para esta classificação está centrado nos mecanismos de coordenação,

caracterizados pela distribuição da decisão e por um maior ou menor esforço nas comunicações.

2.4 Organizações Virtuais

2.4.1 Definições

De forma geral, para a Ciência, ser virtual é ser de uma certa natureza em efeito ou em implicações, embora não sendo literalmente desta mesma natureza [79].

O termo empresa virtual é atribuído a Mowshowitz [80], o qual estabelece uma analogia com o termo memória virtual no âmbito da Ciência da Computação. Davidow e Malone [81] apresentam o termo orientado aos negócios.

Uma organização virtual pode ser definida como um agrupamento implícito de agentes económicos com diferentes competências, que se complementam através da cooperação, para o alcance de um objectivo comum. Estes agentes económicos são organizações, unidades de uma mesma organização ou indivíduos, os quais podem participar de mais do que um agrupamento simultaneamente.

Os agrupamentos apresentam diferentes existências temporais e ocorrem:

(i) para responder a uma oportunidade de negócio. Um agente económico identifica (baseado nos seus planos) uma oportunidade de negócio e passa a desempenhar o papel de organizador [63], buscando colaboradores na sociedade, ou mesmo internamente à organização, para responder à oportunidade identificada, ou

(ii) para encontrar uma oportunidade de negócio aproveitando as competências disponíveis.

Nos processos de reengenharia das organizações, onde o ênfase está colocado nos processos, de forma a que os mesmos apresentem qualidade (características que atendam às expectativas dos consumidores) e rapidez exigida pela competição dos mercados, é usual a formação de equipas temporárias (durante a execução do projecto), denominadas de equipas virtuais [55]. Equipas virtuais são compostas por especialistas de unidades distintas, que uma vez encerrado os processos, que motivaram as suas formações, as competências retornam às suas unidades de origem ou passam a integrar outras equipas.

2.4.2 Cooperação em Estruturas Virtuais

A cooperação dos diferentes agentes económicos (agentes internos à organização interessada na criação de estruturas virtuais ou agentes externos oriundos do mercado) para a criação de uma estrutura virtual ocorre fundamentalmente através da disponibilização de suas competências específicas (capacidades, conhecimento sobre o saber-fazer e tecnologia [82]), não envolvendo as estruturas que os suportam e nem o comportamento que nelas exibem ou a que nelas estão sujeitos.

Para a criação de uma estrutura virtual, duas etapas básicas são necessárias:

(i) a definição do processo a ser executado para o alcance de um objectivo e seu particionamento, e

(ii) a montagem da estrutura para implementar esse processo.

Quanto ao particionamento do processo, a solução é auxiliada pela adopção de um formalismo que permita de maneira uniforme a descrição dos processos parciais. A

metodologia para arquitectura de sistemas integrados de informação - "*ARIS*" ("*Architecture of Integrated Information Systems*") [83] apresenta uma maneira uniforme para a descrição e modelação de processos de negócios (anexo H). A segunda etapa, relativa à formação da estrutura virtual, é composta da identificação e selecção dos parceiros e atribuição dos processos parciais a estes.

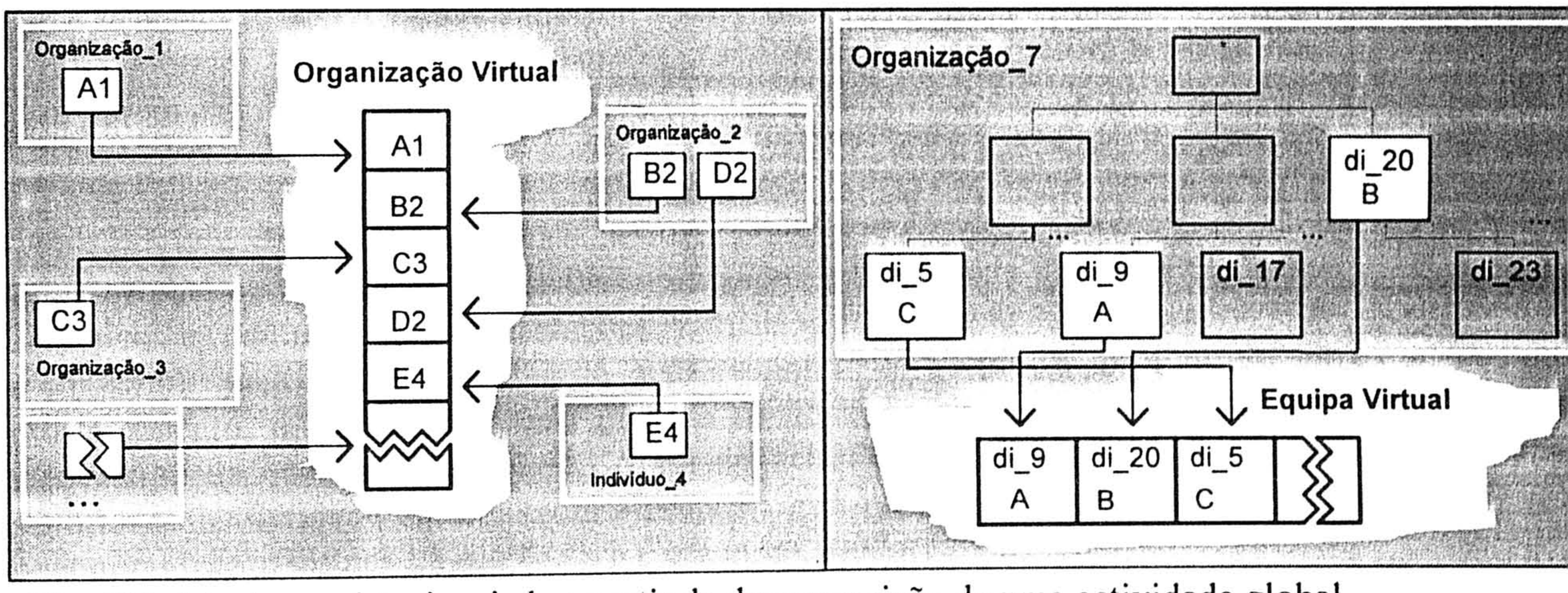
Os mecanismos utilizados para a coordenação das actividades numa organização virtual baseiam-se fundamentalmente:

(i) no ajuste mútuo (através da negociação e da argumentação), e

(ii) no papel de um gerente de produto.

O gerente de produto ou é desempenhado por uma das organizações ou é distribuído pelos diversos parceiros. O gerente de produto é responsável pela coordenação da actividade de selecção de parceiros e de atribuição aos mesmos dos processos parciais (os quais foram decompostos a partir do processo global).

A [fig. 30] ilustra estruturas virtuais criada a partir da decomposição de uma actividade global em "N" partes (A, B, C, D, E, ..., N). No caso da organização virtual, a atribuição das tarefas, após o processo de selecção de parceiros, foi designada para as organizações (1, 2 e 3) e para o indivíduo (4) que cooperarão com suas competências nas respectivas partes do processo global (organização_1 com a competência para a parte A do processo global, organização_2 com as competências para a parte B e D do processo global, etc.). No caso de uma equipa virtual, para as tarefas decompostas, foram designadas as competências das divisões (di_5, di_9 e di_20) da mesma organização (organização_7).



[fig. 30] Estruturas virtuais criada a partir da decomposição de uma actividade global

2.4.3 O Projecto "*AVE*"

Um projecto que modela estruturas virtuais a partir de uma sociedade de agentes inteligentes é o projecto "*AVE*" ("*Agents in Virtual Enterprises*") [84]. Os agentes desta sociedade são baseados na arquitectura "*InteRRaP*" [85]. Esta arquitectura modela um agente através de três componentes:

(i) uma interface com o mundo,

(ii) uma unidade de controlo, e

(iii) uma base de conhecimento.

A interface com o mundo contém sensores, actuadores e é dotada da capacidade de comunicação. A unidade de controlo consiste em três camadas: uma camada de

comportamentos, uma camada de planeamento local e uma de planeamento cooperativo. A base de conhecimento consiste nos modelos: do mundo, mental e social. Na unidade de controlo, a função da camada de comportamentos é a de reagir face a situações consideradas críticas. Reacções são disparadas por eventos reconhecidos, contidos no modelo do mundo. A camada de planeamento local capacita a deliberação do agente através da construção do modelo de informação do mundo, utilizando os actuais objectivos do agente e as intenções locais mantidas no modelo mental, bem como os mecanismos disponíveis para o planeamento. A camada do planeamento cooperativo estende as funcionalidades de planeamento do agente para cobrir a elaboração de planos conjuntos, i.e. o planeamento com outros agentes e a resolução de conflitos. Esta camada utiliza informação sobre os objectivos dos outros agentes, suas capacidades e compromissos, os quais estão armazenados no modelo social.

A comunidade de agentes no projecto "*AVE*" inicia suas actividades a partir de um processo global de um dos agentes, o qual é descrito utilizando a metodologia "*ARIS*". Este processo é decomposto numa sequência de sub-processos baseada nos modelos clássicos de organizações, i.e. compras, produção, marketing, vendas e distribuição. Para instanciar os agentes económicos que participarão da estrutura virtual, o processo de leilões é utilizado. Os processos de **leilões** podem ser abertos ou fechados. São abertos quando as ofertas são efectuadas publicamente e os participantes conhecem as ofertas uns dos outros. São fechados quando as ofertas são conhecidas apenas pelo leiloeiro. Os leilões fechados podem ser ainda diferenciados pelos de primeiro preço e segundo preço. Nos de primeiro preço o leiloeiro atribui como vencedor o ofertante que anunciou a mais alta oferta, tendo o mesmo que pagar o valor ofertado. Já no de segundo preço, o vencedor é o que ofertou o preço mais alto, porém pagará o segundo mais alto valor oferecido.

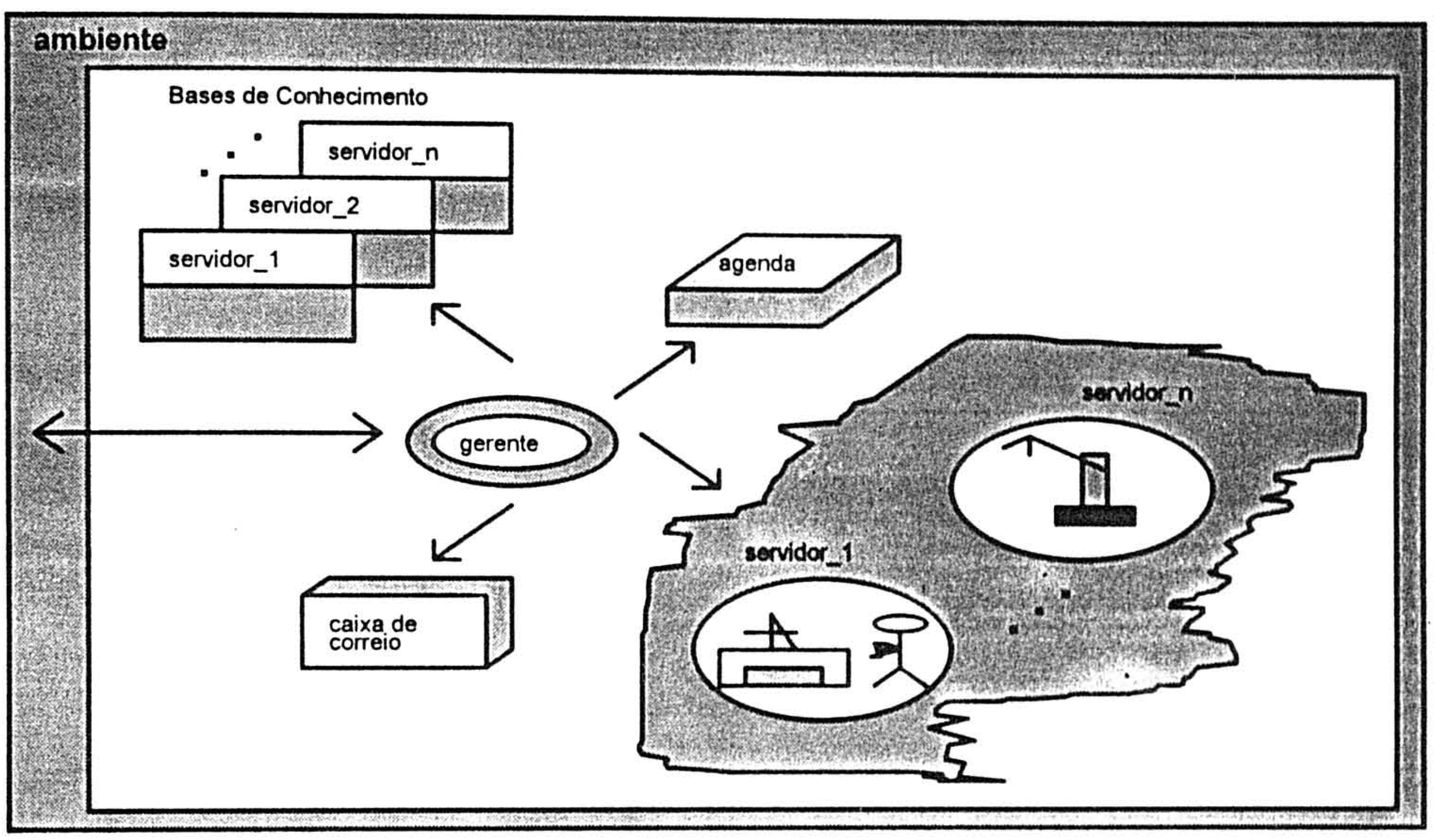
2.4.4 O Modelo de Rabelo e Camarinha-Matos

Rabelo e Camarinha-Matos [86] apresentam um ambiente para escalonamento dinâmico e controlo em organizações virtuais de produção de recursos. Uma arquitectura multiagente é proposta para suportar a gestão e o controlo do escalonamento, o qual é executado por um conjunto de agentes distribuídos, cujo mecanismo de coordenação é baseado em negociação. Um dos fundamentos para suportar a flexibilidade da arquitectura proposta está relacionado com a adopção de uma organização virtual, cujo objectivo é orientar a produção à tarefas.

O sistema de escalonamento modela os recursos em duas camadas: a física e a virtual. Na camada física os recursos de produção ,são estáticos de acordo com a planta de fabrico instalada. Na camada virtual estes recursos de produção são vistos como um agrupamento lógico dos mesmos, os quais podem ser de dois níveis: o nível virtual estático e o nível virtual dinâmico. O nível virtual estático representa uma primeira estruturação virtual da camada física, cuja topologia é organizada em termos de funcionalidades dos recursos de produção. O nível virtual dinâmico representa uma estrutura virtual a partir da estrutura virtual estática, através da formação de um consórcio. Um consórcio representa um escalonamento descentralizado, e é um lógico, dinâmico e temporário agrupamento de recursos de produção para a execução de uma tarefa.

Para suportar o sistema multiagente citado, existem quatro tipos de agentes [87]:

(i) agente de actividade empresarial ("*enterprise activity agent*"), para o nível físico [fig. 31],
(ii) agente de centro de expansão local ("*local spreading center agent*") para o nível virtual estático,
(iii) agente de consórcio ("*consortium agent*") para o nível virtual dinâmico, e
(iv) agente supervisor do escalonamento ("*scheduling supervisor agent*"), o qual é responsável pelo controlo e pela gestão global do escalonamento, desempenhando um papel de controlador.



[fig. 31] Agente de actividade empresarial no modelo de Rabelo e Camarinha-Matos

O agente de actividade empresarial é composto por:
(i) servidores os quais representam os recursos que estão funcional e fisicamente ligados uns com os outros, compondo uma equipa,
(ii) o gerente, o qual é responsável pela coordenação e pela representação da equipa face aos demais agentes da comunidade,
(iii) uma agenda, a qual contém a sequência de actividades empresariais contratadas para serem executadas,
(iv) uma caixa de correio, a qual contém o registo das actividades empresariais em processo de negociação, e
(v) uma base de conhecimento, contendo a descrição (capacidades, características topológicas e informação de gestão) de todos os servidores pertencentes a este agente.

2.5 Organizações Artificiais - O sistema "*ARTOR*"

Intimamente ligado com o tornar virtuais as Organizações está a progressiva tendência de torna-las artificiais.

Atribuir às organizações um maior coeficiente de artificialidade (a razão entre o número de processos executados por artefactos computacionais e a totalidade de processos de uma organização), significa incrementar o "*inner*" das organizações com artefactos que passem a executar parte do que era até então executado pelos recursos-humanos de uma organização. Certas relações (passivas ou activas) entre o "*inner*" e o "*outer*" ou mesmo as intra-"*inner*" passam a não mais ocorrer apenas entre seres humanos, mas também entre seres humanos e artefactos e entre artefactos e artefactos.

O sistema "*ARTOR*" ("*ARTificial ORganizations*") é um sistema computacional baseado em metodologias de Inteligência Artificial Distribuída e Aprendizagem Simbólica Automática, cujo objectivo é, de forma incremental, construir e manter um ambiente que permita simular um cenário de uma sociedade de organizações nas múltiplas dimensões intra e interorganizações.

Este ambiente tem tido seu desenvolvimento baseado na programação em lógica na sua versão "*YAP*" ("*Yet Another Prolog*") [92]. As comunicações são suportadas por rotinas escritas em linguagem "*C*", utilizando rotinas da biblioteca "*TCP/IP*" de manipulação de "*sockets*", as quais estão integradas através de predicados (de criação de "*sockets*", leitura e de gravação de mensagens) na versão do "*YAP*". As estações de trabalho utilizadas são "*SUN*" e "*DEC*", porém de fácil transposição para outras máquinas que suportam a arquitectura "*Unix*".

2.5.1 Arquitectura do sistema "*ARTOR*"

Dos componentes activos de um cenário representativo de uma sociedade de organizações (organizações e seres humanos como indivíduos) apenas as organizações, em consequência seus "*inners*", se tornam artefactos. Os artefactos que modelam as organizações são agentes denominados de agentes envolventes ("*cover agents*"). Os artefactos que modelam o "*inner*" de uma organização são: agentes gestores ("*administrator agents*"), agentes executores ("*executor agents*") e uma base de conhecimento corporativo ("*corporate knowledge base*").

Na arquitectura do "*ARTOR*", cada organização é um agente envolvente, o qual:

(i) quanto à sua constituição:
  (i.i) de origem é artificial,
  (i.ii) de existência é virtual, e
  (i.iii) de granularidade é super,
(ii) quanto às suas capacidades e comportamentos é:
  (ii.i) cognitivo,
  (ii.ii) semi-autónomo, devido à necessidade de estabelecer relações (activas) com outros agentes envolventes (organizações) ou com indivíduos, e
  (ii.iii) social.

As relações (passivas e activas) que suportam o contacto entre os componentes activos da sociedade de organizações podem ocorrer de três formas:

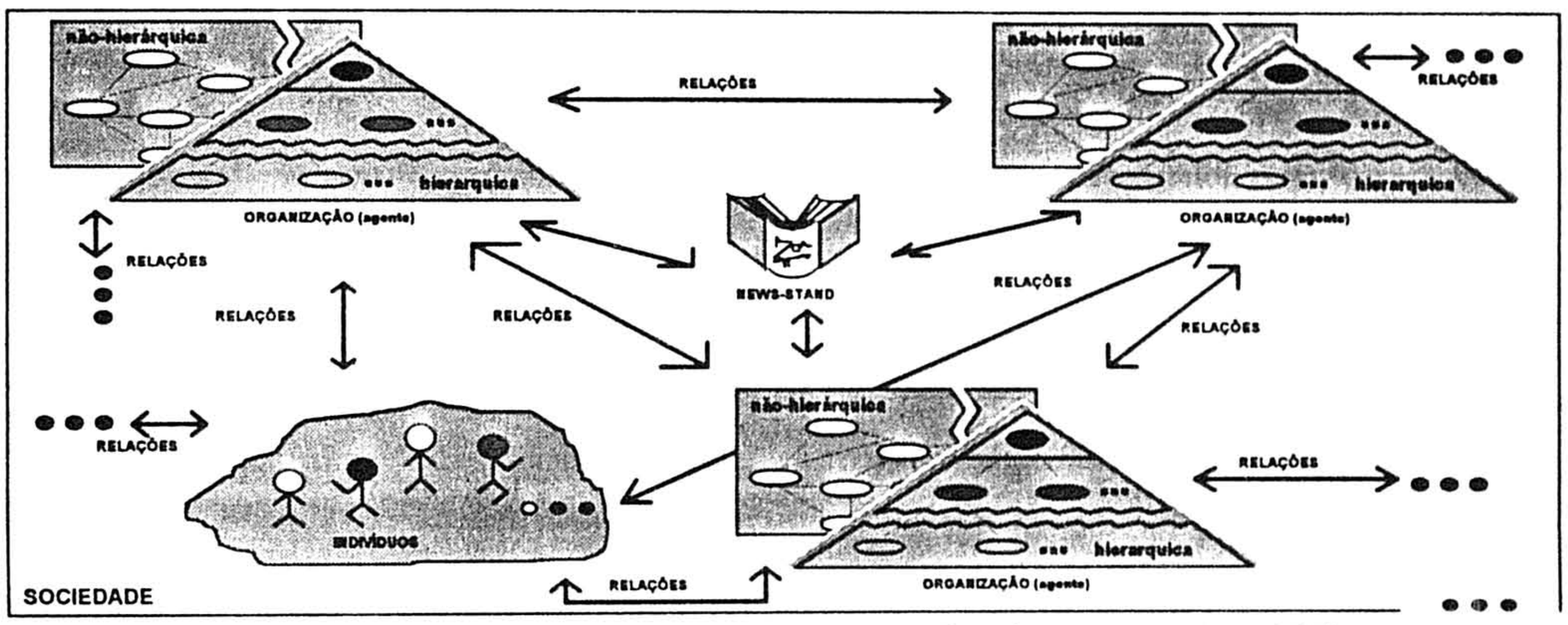
(i) **ponto a ponto** quando um componente activo endereça a sua mensagem ou observação directamente a outro componente activo,

(ii) **ponto a multi-ponto** quando um componente activo endereça para mais de um componente activo, a sua mensagem ou observação, e

(iii) utilizando o "***news-stand***" (quiosque), quando um componente activo deseja divulgar notícias à sociedade. Para este caso, um "quadro-negro" denominado de "*news-stand*" é introduzido, o qual recebe as informações dos agentes, disponibilizando-as aos demais agentes da sociedade interessados na informação. Entende-se por agentes interessados aqueles que, baseados em seus planos, procuram o conhecimento para gerar ou aproveitar uma oportunidade de negócio. O "*news-stand*" controla os acessos através de datas limites para a veiculação da notícia.

Os indivíduos estabelecem suas relações com os artefactos através de uma interface homem-artefacto.

Do ponto de vista da sociedade, os componentes activos da mesma são distribuídos, descentralizados, agindo segundo as suas capacidades cognitivas. Eles são homogéneos se pertencerem a uma mesma especialidade. Caso contrário são heterogéneos. A [fig. 32] representa a sociedade modelada pelo sistema "*ARTOR*".



[fig. 32] - A Sociedade modelada pelo sistema "*ARTOR*"

2.5.2 Definição do sistema "*ARTOR*"

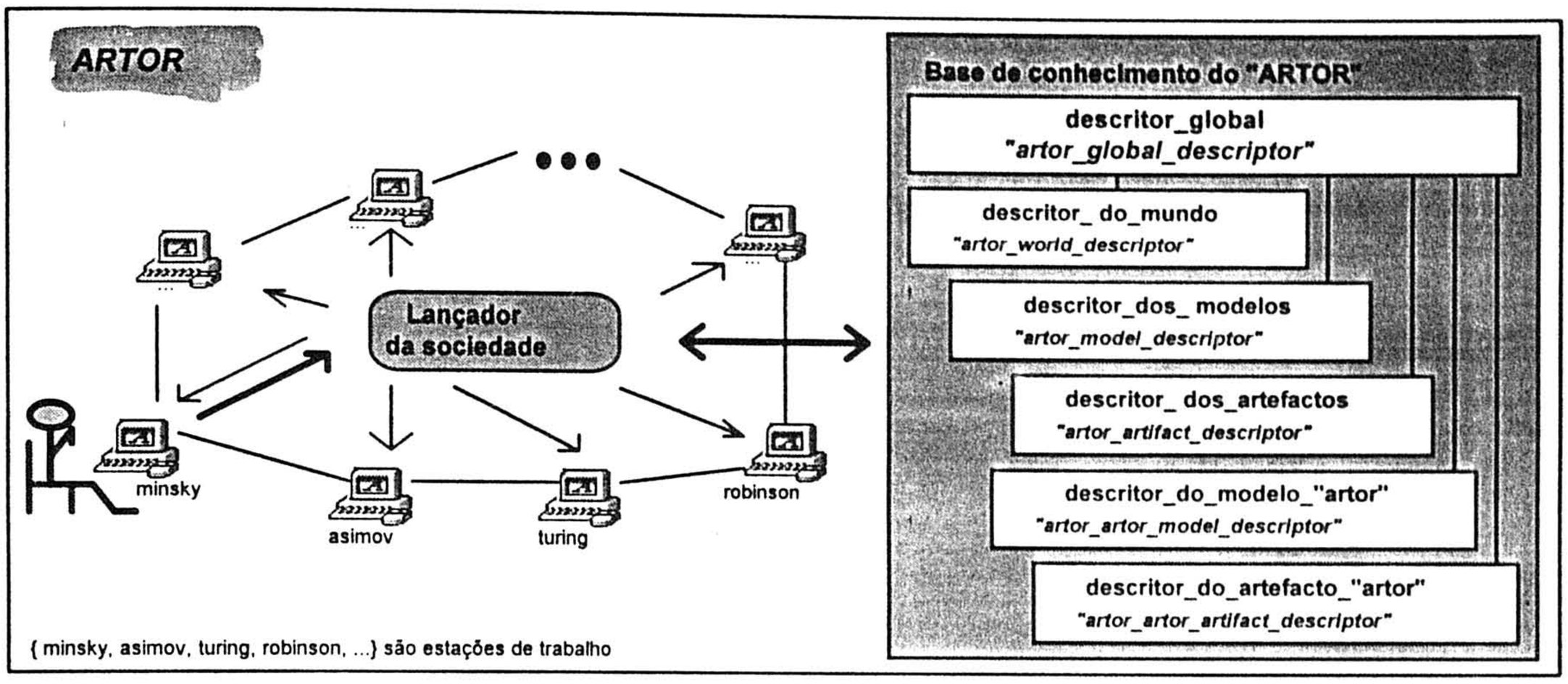
O sistema "*ARTOR*" é, ao nível mais abstracto, definido pelo par ordenado:

> (***Ms***, ***Ls***) onde:
> (i) ***Ms*** é a base de conhecimento, e
> (ii) ***Ls*** é o lançador da sociedade.

O base de conhecimento ($_{Ms}$) contém o conhecimento simbólico no domínio do sistema, representado através de redes semânticas, estruturado em seis módulos denominados de descritores [fig. 33]:

(i) global,
(ii) do mundo,
(iii) dos artefactos,
(iv) do modelo "*ARTOR*", e
(v) do artefacto "*ARTOR*".

Seguidamente apresentam-se as instâncias dos predicados para cada descritor. Cada um dos predicados tem como primeiro argumento o identificador do nível de conhecimento a que se refere (é dado em cada caso o nome do predicado descrito e o número de argumentos).



[fig. 33] O sistema "*ARTOR*" ao nível mais abstracto

(i) o **descritor_global** ("*artor_global_descriptor*") descreve o modelo global do conhecimento, i.e. o meta conhecimento sobre o sistema "*ARTOR*" [fig. 34], através dos predicados:

is_part_of/3
módulos de conhecimento que fazem parte do modelo de conhecimento do sistema "*ARTOR*",

describes/3
o significado de cada módulo, e

is_located/5
localização dos módulos de conhecimento nas estações de trabalho. Respectivamente correspondem: (i) ao identificador do conhecimento, (ii) tipo de módulo, (iii) a identificação do módulo de conhecimento, (iv) a estação de trabalho, e (v) o endereço de localização na respectiva estação.

```
% i_am( artor,  artor_global_descriptor).
%
is_part_of( artor, artor_world_descriptor, artor_global_descriptor).
is_part_of( artor, artor_model_descriptor, artor_global_descriptor).
...
%
describes( artor, artor_world_descriptor, world).
describes( artor, artor_model_descriptor, model).
...
%
is_located( artor,  descriptor, artor_world_descriptor, minsky,  '/usr/users/shmeil/ARTOR.dir').
...
```

[fig. 34] O descritor global do sistema "*ARTOR*"

(ii) o **descritor_do_mundo** ("*artor_world_descriptor*") descreve o modelo conceptual do mundo [fig. 35], através dos predicados:

is_a/3,
is_part_of/3

```
% i_am( world, artor_world_descriptor).
%
is_a( world, real_world, world).
human_being).
%
human_being).
is_part_of( world, natural_world, real_world).
is_part_of( world, artificial_world, real_world).
is_part_of( world, natural_object, natural_world).
is_part_of( world, natural_phenomena, natural_world).
is_part_of( world, artifact, artificial_world).
%
o  is_a( world, human_being, natural_objects).
o  is_a( world, other_natural_object, natural_object).
o  is_part_of( world, cognitive_capability,
o  is_part_of( world, physical_capability,
o  is_part_of( world, perception, cognitive_capability).
o  ...
o  %
o  is_part_of( world, action, physical_capability).
o  ...
o
```

[fig. 35] O descritor do mundo

(iii) o **descritor_dos_modelos** (“*artor_model_descriptor*”) descreve o modelo conceptual dos modelos [fig. 36], através dos predicados:

is_a/3,
is_part_of/3

```
% i_am( model, artor_model_descriptor).
%
hierarchical_organization).
is_a( model, society, model).
is_a( model, organization, model).
is_a( model, artor_model, model).
is_a( model, other_model, model).
...
%
is_a( model, society_of_organizations, society).
hierarchical_layer).
is_a( model, other_society, society).
...
%
is_part_of( model, human_being, society_of_organizations).
is_part_of( model, organization, society_of_organizations).
is_part_of( model, social_relation, society_of_organizations).
...
%
is_a( model, hierarchical_organization, organization).
is_a( model, other_organization, organization).
...
o  is_part_of( model, corporate_knowledge, organization).
o  is_part_of( model, hierarchical_layer,
o  %
o  is_part_of( model, managing_knowledge_capability,
o  corporate_knowledge).
o  ...
o  %
o  is_a( model, top_layer, hierarchical_layer).
o  is_a( model, medium_vertical_horizontal_layer,
o  is_a( model, low_horizontal_layer, hierarchical_layer).
o  %
o  is_part_of( model, medium_vertical_horizontal_layer, top_layer).
o  is_part_of( model, low_horizontal_layer,
o  medium_vertical_horizontal_layer).
o  ...
o  %
o  is_part_of( model, managing_capability, top_layer).
o  is_part_of( model, learning_capability, top_layer).
o  ...
o
```

[fig. 36] O descritor dos modelos

(iv) **descritor_dos_artefactos** (“*artor_artifact_descriptor*”) descreve o modelo dos artefactos (“*software*” e “*hardware*”) para desenvolvimento e execução do sistema [fig. 37], através dos predicados:

is_a/3,
is_part_of/3,
is_located/5

```
% i_am( artifact, artor_artifact_descriptor).
%
is_a( artifact, hardware_artifact, artifact).
is_a( artifact, software_artifact, artifact).
is_a( artifact, other_artifact, artifact).
software_artifact).
is_a( artifact, artor_software, software_artifact).
is_a( artifact, network, hardware_artifact).
is_a( artifact, workstation, hardware_artifact).
is_a( artifact, hard_disk, hardware_artifact).
is_a( artifact, other_hardware, hardware_artifact).
o  is_a( artifact, niar_local_area_network, local_area_network ).
o  %  niar (nucleo de inteligencia artificial e robotica).
o  ...
o  %
o  is_a( artifact, application_development_software,
o  %
o  is_part_of( artifact, program_development,
o  application_development_software).
o  is_part_of( artifact, routine_development,
o  application_development_software).
```

```
is_a( artifact, local_area_network, network).
is_a( artifact, other_network, network).
...
%
```

```
is_a( artifact, yap, program_development).
...
%
is_located( artifact,  program_development, yap, minsky,
                '/usr/users/shmeil/meu_wyapn').
```

[fig. 37] O descritor dos artefactos

(v) **descritor_do_modelo_"*artor*"** ("*artor_artor_model_descriptor*") descreve o modelo conceptual dos elementos da sociedade suportado pelo sistema "*ARTOR*" [fig. 38], através dos predicados:

is_a/3,
is_part_of/3,
represents/3

```
% i_am( artor_model,  artor_artor_model_descriptor).
%
is_part_of( artor_model, cover_agent, artor_model).
is_part_of( artor_model, news_stand,  artor_model).
is_part_of( artor_model,
            artor_human_being_machine_interface,
            artor_model).
%
adminstrator_agent).
is_a( artor_model, hierarchical_cover_agent,
            cover_agent)).
is_part_of( artor_model,
            corporate_knowledge_data_base,
            hierarchical_cover_agent).
is_part_of( artor_model, agent_hierarchical_layer,
            hierarchical_cover_agent).
managing_capacity).
%
managing_capacity).
is_part_of( artor_model,
            managing_knowledge_capacity,
            corporate_knowledge_data_base).
is_part_of( artor_model, communication_capacity,
            corporate_knowledge_data_base).
%
```

```
is_a( artor_model, agent_top_layer,
            agent_hierarchical_layer).
...
%
is_part_of( artor_model, administrator_agent, agent_top_layer).
...
%
is_part_of(          artor_model,          managing_capacity,

is_part_of( artor_model, learning_capacity, adminstrator_agent).
is_part_of( artor_model, communication_capacity,
            adminstrator_agent).
is_part_of( artor_model, individual_knowledge_data_base,
            adminstrator_agent).
%
is_part_of(          artor_model,          planning_capacity,

is_part_of(          artor_model,          coordination_capacity,

...
%
represents( artor_model, artor, society_of_organizations).
...
```

[fig. 38] O descritor do modelo "*artor*"

(vi)**descritor_do_artefacto_"*artor*"** ("*artor_artor_artifact_descriptor*") descreve o modelo instanciado dos artefactos que fazem parte de uma execução do sistema "*ARTOR*" [fig. 39], através dos predicados:

is_a/3,
is_competent_start/3,
has_environmental_attribute/5

```
% i_am( artor_artifact,  artor_artor_artifact_descriptor).
%
is_a( artor_artifact, artor_menus1, artor_human_being_machine_interface_choice).
is_a( artor_artifact, artor_society_X_ray1, artor_human_being_machine_interface_status).
is_a( artor_artifact, artor_stand_news1, news_stand).
is_a( artor_artifact, artor_org_A, cover_agent).
is_a( artor_artifact, artor_org_1, cover_agent).
is_a( artor_artifact, artor_org_2, cover_agent).
...
%          These following clauses specify the competency to start an artor_component
%
is_competent_start( artor_artifact, User, 'ARTOR').
is_competent_start( artor_artifact, 'ARTOR', artor_human_being_machine_interface_choice).
is_competent_start( artor_artifact, artor_human_being_machine_interface_choice,  artor_human_being_machine_interface_status).
is_competent_start(artor_artifact, artor_human_being_machine_interface_choice, news_stand).
```

```
is_competent_start( artor_artifact, artor_human_being_machine_interface_choice, cover_agent).
%
%          These following clauses specify some artor's instance environmental attributes to artor_artifact to continue
%                    has_environmental_att(  artor_artifact, Instance_name, Workstation_name, Port_number, [X, Y, W, H, C])
%                    where: X, Y are coordinates, W is width, H is height and C is color for human_being_machine_interface
%
%          Remarks: when the Workstation_name = _,  means the artor_element will be start in the
%                                                   same workstation that the ARTOR was.
%
has_environmental_att( artor_artifact, artor_menus1, _, _, []).
has_environmental_att( artor_artifact, artor_society_X_ray1, minsky, 3000, [845, 832, 370, 180, _]).
has_environmental_att( artor_artifact, artor_stand_news1, minsky, 3050, [2, 870, 700, 140, yellow1]).
has_environmental_att( artor_artifact, artor_org_A, turing, 3600, []).
has_environmental_att( artor_artifact, artor_org_1, asimov, 3300, []).
has_environmental_att( artor_artifact, artor_org_2, robinson, 3400, []).
...
```

[fig. 39] O descritor do artefacto "*artor*"

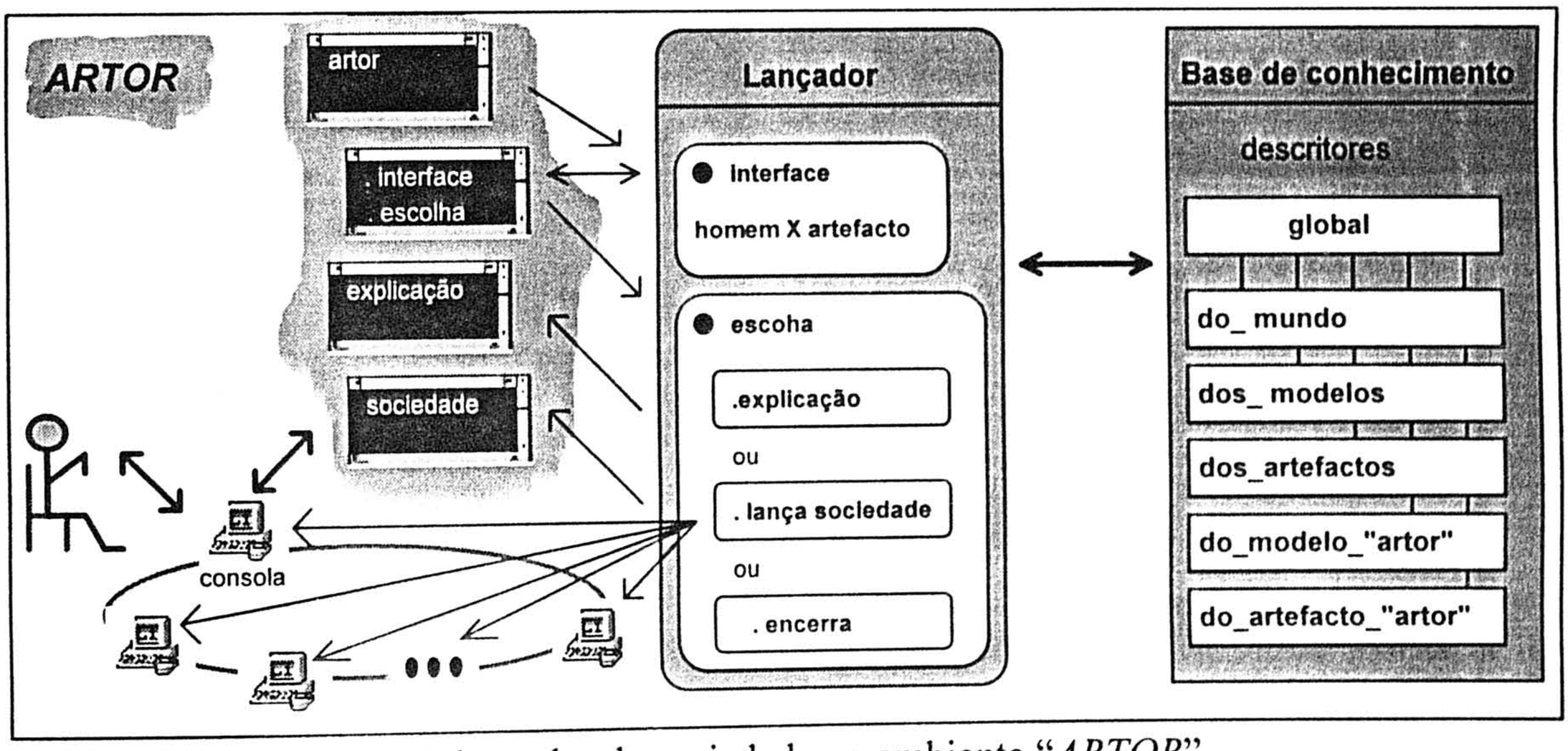
O conhecimento contido nos descritores, que fazem parte da base de conhecimento, apresenta duas utilidades:

(i) apoiar as explicações sobre o sistema "*ARTOR*", e

(ii) permitir estruturar e dinamicamente carregar o ambiente (conceitos, "*software*", "*hardware*" e a configuração da sociedade) para uma execução.

O conhecimento fornecido pelos descritores estrutura a sociedade, não cobrindo o conhecimento sobre o "*inner*" dos componentes. O conhecimento estrutural do "*inner*" dos componentes está contido na base de conhecimento de cada um deles ou seja, a sociedade desconhece o interior dos componentes, há apenas uma visão social dos mesmos.

O lançador da sociedade [fig. 40], por sua vez, baseado no conhecimento contido nos descritores, lança a interface homem-artefacto.



[fig. 40] O lançador da sociedade no ambiente "*ARTOR*"

Esta interface habilita o utilizador escolher entre:

(i) explicações sobre o sistema ("*What About*"),

(ii) prosseguimento e apresentação dos componentes da sociedade para uma execução específica ("*Go Ahead*"), e

(iii) encerramento da sessão ("*Exit*").

Os itens a seguir, descrevem as acções dos componentes e os recursos utilizados em função da escolha efectuada:

(i) a partir da digitação, na estação de trabalho escolhida como consola, da palavra "*ARTOR*", é apresentado o ecrã geral do sistema [fig. 41].

Para este fim, os descritores utilizados são: o global, o artefacto e o artefacto_"*artor*". Do descritor_global é obtido o endereço dos descritores: artefacto e artefacto_"*artor*". Por exemplo, se a estação de trabalho escolhida como consola for a chamada "*minsky*", teremos:

```
is_located(                    artor,                    descriptor,
artor_artor_artefact_descriptor,              minsky,
'/usr/users/shmeil/ARTOR.dir').
is_located( artor, descriptor, artor_artefact_descriptor,
                    minsky, '/usr/users/shmeil/ARTOR.dir').
```

A identificação da estação de trabalho é obtida a partir do "*Unix*" através do comando "*hostname*".

Dos descritores, artefacto e artefacto_"*artor*" são obtidas: a competência para o lançamento dos componentes e as instâncias dos componentes, possibilitando uma reconfiguração do sistema (experiências com novas versões dos componentes), sem alteração na estrutura de controlo.

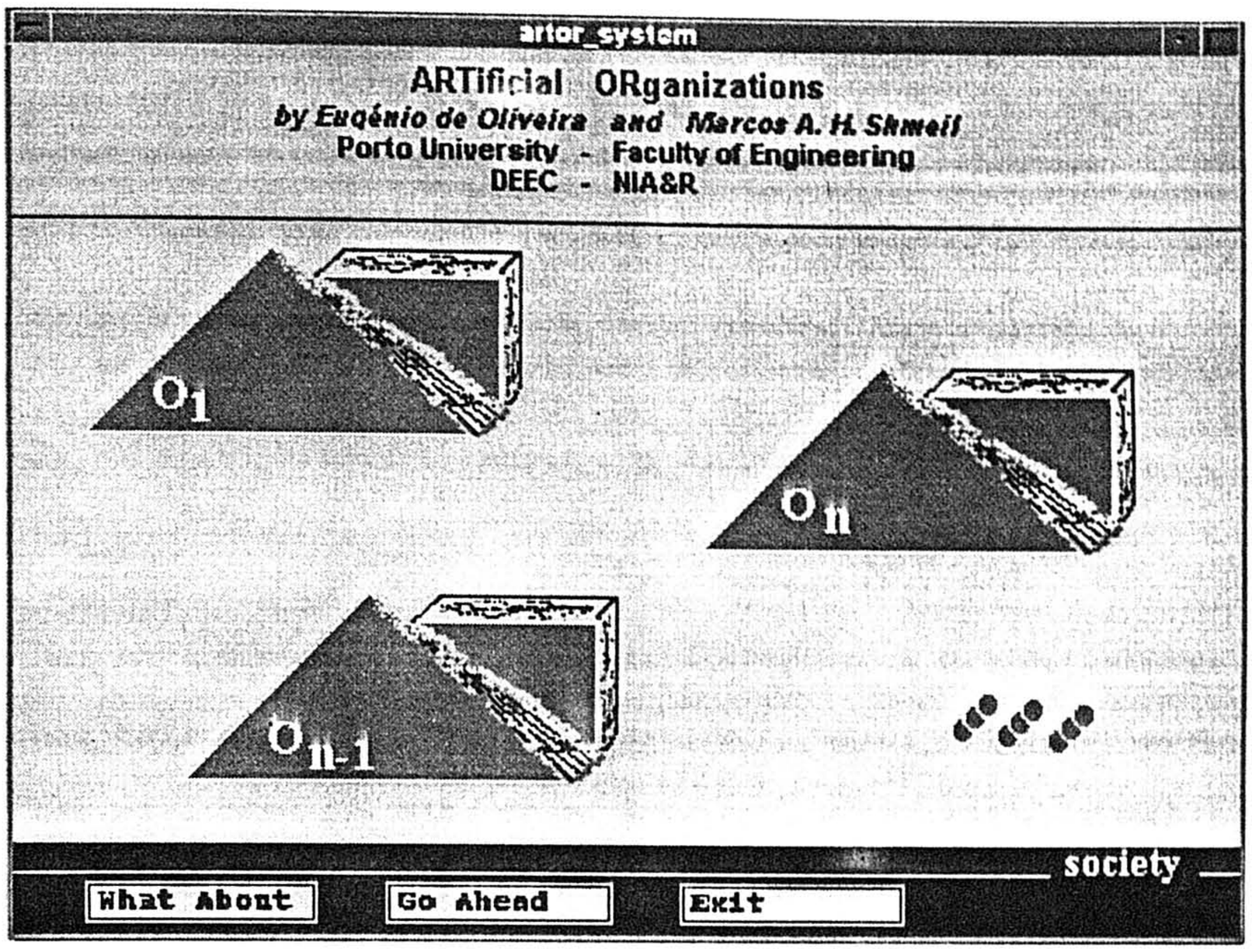
O predicado "*is_competent_start*" orienta a sequência de execução, i.e. o utilizador ("*User*") dispara o "*ARTOR*" e o "*ARTOR*" é competente para disparar a interface homem-artefacto ("*artor_human_being_machine_interface_choice*"), a qual neste exemplo é suportada pelo programa "*artor_menus1*".

```
is_a( artifact, artor_menu1, artor_program).
is_located( artifact,  artor_program, artor_menu1, minsky,
                    '/usr/users/shmeil/ARTOR.dir').


is_a( artor_artifact, artor_menus1,
              artor_human_being_machine_interface_choice).
has_environmental_att(  artor_artifact,  artor_menus1,  _,  _,
[]).

is_competent_start( artor_artifact, User, 'ARTOR').
is_competent_start( artor_artifact, 'ARTOR',

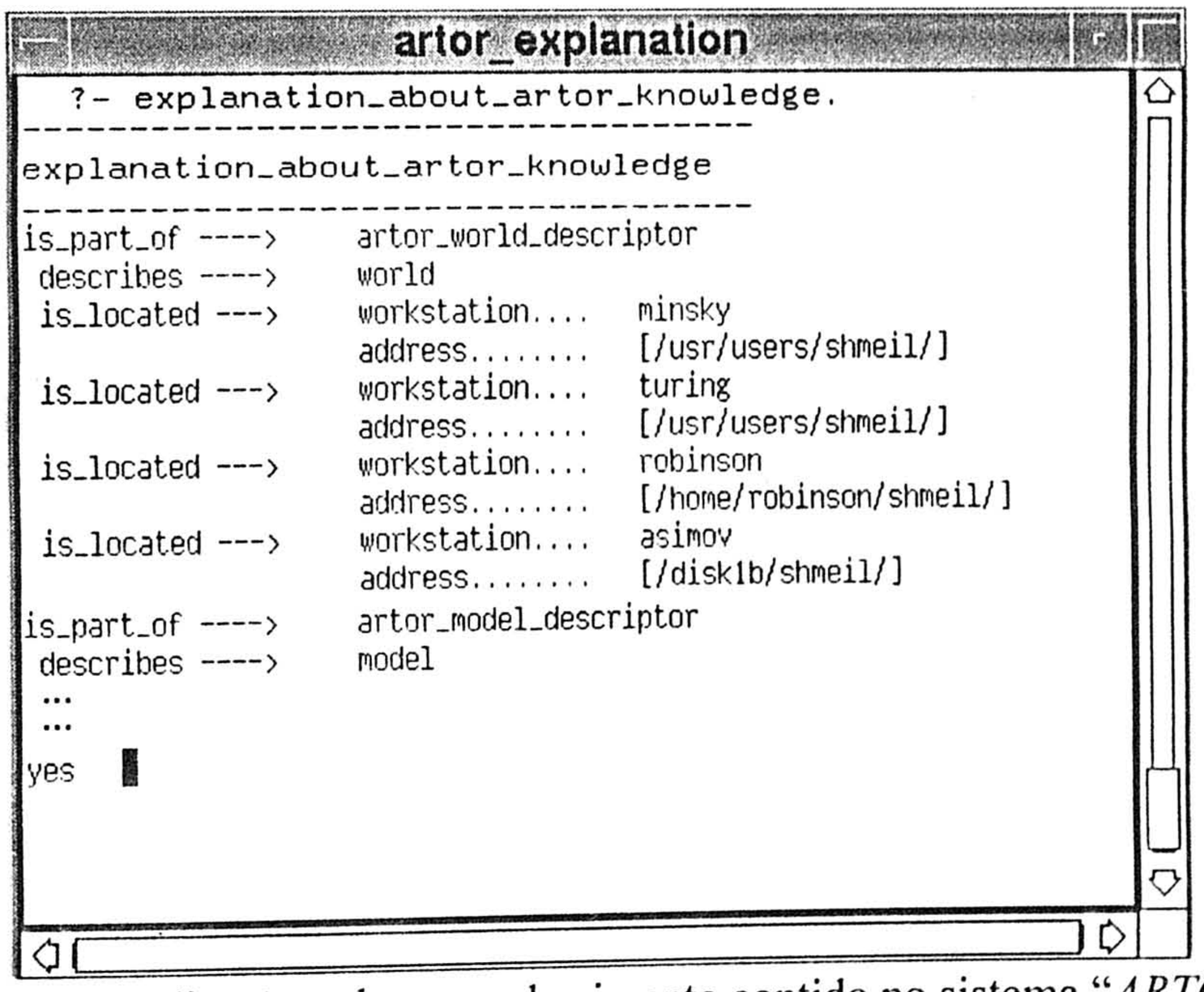
artor_human_being_machine_interface_choice).
```



[fig. 41] Ecrã geral do sistema "*ARTOR*"

(ii) Posteriormente, se a escolha for a de fornecer explicação ("*What About*") sobre o sistema, esta possibilita obter o conhecimento contido nos vários descritores, em função da explicação desejada.

Por exemplo [fig. 42] o "*explanation_about_artor_knowledge*" explica sobre a composição da base de conhecimento do sistema.



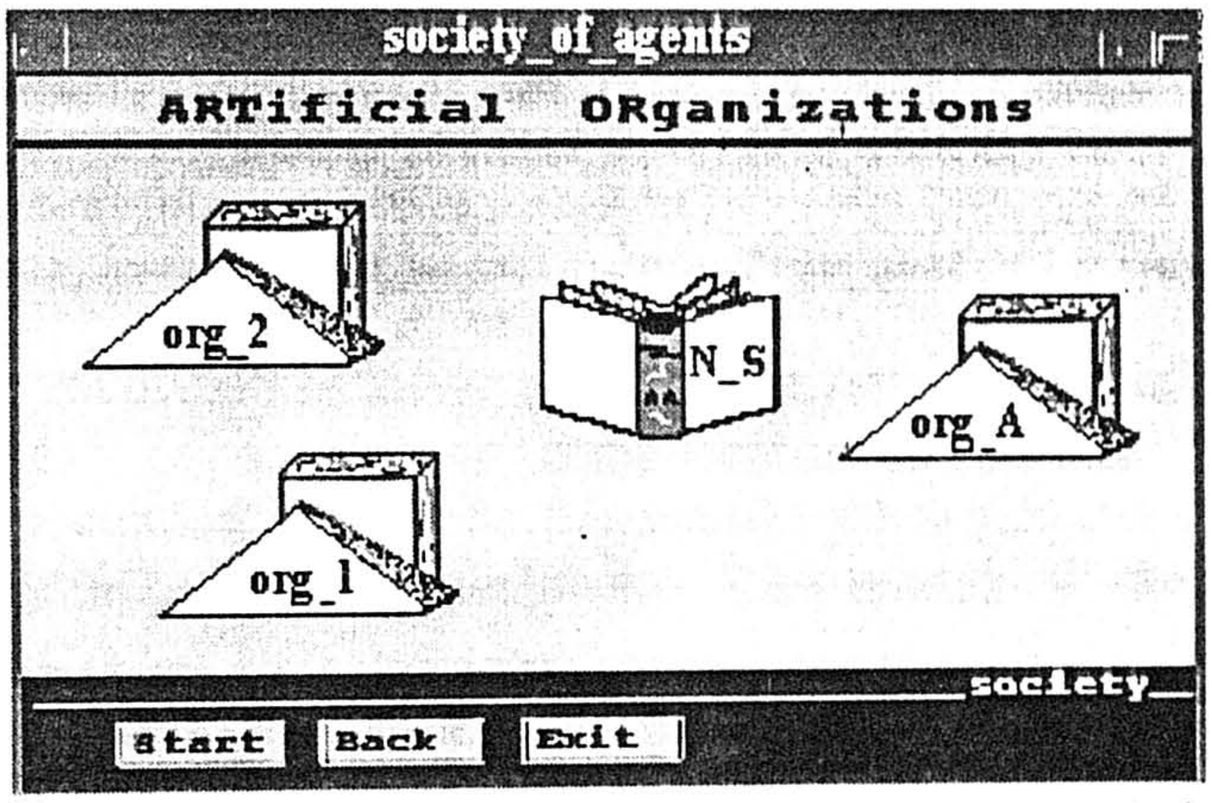
[fig. 42] Explicação sobre o conhecimento contido no sistema "*ARTOR*"

(iii) Se a opção for a de dar prosseguimento ao lançamento da sociedade ("*Go Ahead*"), o ecrã [fig. 43] contendo a configuração instanciada para a específica execução é apresentado. Esta opção

utiliza o conhecimento contido no descritor_artefacto_"*artor*", o qual para este exemplo contém quatro componentes, os quais serão suportados por estações de trabalho distintas (Minsky, Turing, Asimov e Robinson).

```
is_a( artor_artifact, artor_stand_news1, news_stand).
is_a( artor_artifact, artor_org_A, cover_agent).
is_a( artor_artifact, artor_org_1, cover_agent).
is_a( artor_artifact, artor_org_2, cover_agent).

has_environmental_att( artor_artifact, artor_stand_news1,
                       minsky, 3050, [2, 870, 700, 140,
yellow1]).
has_environmental_att( artor_artifact, artor_org_A, turing,
                       3600, []).
has_environmental_att( artor_artifact, artor_org_1, asimov,
                       3300, []).
has_environmental_att( artor_artifact, artor_org_2, robinson,
                       3400, []).
```



[fig. 43] Prosseguimento no lançamento da sociedade

O ecrã de prosseguimento no lançamento da sociedade [fig. 43] apresenta três opções:

(i) retornar ao ecrã anterior ("*Back*"),

(ii) encerrar a sessão ("*Exit*"), e

(iii) lançar os componentes da sociedade ("*Start*").

Quando da escolha do lançamento dos componentes da sociedade ("*Start*"), o seguinte procedimento é executado:

(i) com o conhecimento contido no descritor_artefacto_"*artor*", é criado pelo lançador da sociedade (cujo controlo se encontra no módulo de interface) o comando "*Unix*":

"*xon, Estação_de_trabalho, 'ARTOR_start_society*'"

para cada uma das estações de trabalho descritas. Cada uma das estações de trabalho que receberam a mensagem, passam a executar o ficheiro "*ARTOR_start_society*", o qual é um módulo executável,

que lança uma cópia do "*YAP*" para cada um dos componentes da sociedade que foram nomeados pelo descritor_artefacto_"*artor*".
São então lançados:

(i.i) as organizações ("*cover agents*"),

(i.ii) o quiosque ("*news_stand*"), cuja versão neste exemplo é representada pelo programa "*artor_stand_news1*", e

(i.iii) uma interface ("*artor_human_being_machine_interface_status*"), a qual habilita a consulta do conhecimento de cada um dos componentes activos. No exemplo, esta interface é executada pelo programa "*artor_society_X_ray1*".

```
is_a( artor_artifact, artor_society_X_ray1,
            artor_human_being_machine_interface_status).

is_competent_start(artor_artifact,
     artor_human_being_machine_interface_choice,
          artor_human_being_machine_interface_status).
is_competent_start( artor_artifact,
     artor_human_being_machine_interface_choice, news_stand).
is_competent_start( artor_artifact,
     artor_human_being_machine_interface_choice, cover_agent).
```

Os agentes envolventes ("*cover agents*") que representam as organizações (os quais serão tratados na secção: os componentes do sistema "*ARTOR*" - As Organizações), são criados a partir de um modelo de arquitectura genérica, tendo como diferenciador a base de conhecimento. Este modelo é chamado de agente envolvente padrão ("*artor_standard_cover_agent*"), cuja função é carregar a base de conhecimento corporativa e a partir desta providenciar o lançamento dos demais agentes que constituem a organização.

```
is_a( artifact, software_artifact, artifact).
is_a( artifact, artor_software, software_artifact).
is_part_of( artifact, artor_routine, artor_software).
is_a( artifact, artor_standard_cover_agent,
            artor_routine).
is_located( artifact, artor_routine,
                  artor_standard_cover_agent, minsky,
            ['/usr/users/shmeil/ARTOR.dir']).

...
```

2.5.3 Os componentes do sistema "*ARTOR*" - As Organizações



Como referido anteriormente, um dos componentes suportado pelo sistema "*ARTOR*" são as organizações, as quais para o modelo tornam-se agentes envolventes ("*cover agents*") [93]. Os agentes envolventes serão abordados em duas perspectivas, a primeira sob o ponto de vista conceptual organizacional e a segunda sob o ponto de vista computacional.

2.5.3.1 Perspectiva Conceptual Organizacional para um Agente Envolvente

Um **agente envolvente**, do ponto de vista conceptual organizacional, é definido pelo par ordenado:

> (***Ckb***, ***Ags***) onde:
> (i) ***Ckb*** é uma base de conhecimento corporativa, e
> (ii) ***Ags*** é um conjunto de "n" agentes = $\{n : n \in N, n \geq 2\}$.

Deve-se observar que:

(i) os elementos do par ordenado, ou seja ***Ckb*** e ***Ags*** constituem a competência do agente envolvente, e
(ii) os elementos de ***Ags*** são estruturados e estão comprometidos no alcance de objectivos comuns.

Os elementos pertencentes ao conjunto ***Ags*** formam dois sub-conjuntos disjuntos de agentes definidos como segue:

(i) ***Aas***, o sub-conjunto dos agentes gestores ("*administrator agents*"),
e
(ii) ***Eas*** o sub-conjunto dos agentes executores ("*executor agents*").

Observa-se que, pela própria definição, ***Aas*** $\subset$ ***Ags*** e ***Eas*** $\subset$ ***Ags***.

> Sejam ***Aas*** e ***Eas***.
> Uma **organização** é uma tripla (***Aas***, ***Eas***, ***Ckb***), ***Aas*** $\neq \emptyset$, ***Eas*** $\neq \emptyset$, ***Ckb*** $\neq \emptyset$.

Um agente **gestor** é definido pelo par ordenado:

> (***Ikb***, ***Caa***) onde:
> *(i)* ***Ikb*** é uma base de conhecimento individual, e
> (ii) ***Caa*** é um conjunto de capacidades $\equiv$ {planeamento, coordenação, comunicação, aprendizagem}.

Deve-se observar que:

(i) os elementos do par ordenado, ou seja ***Ikb*** e ***Caa*** constituem a competência do agente gestor.

Um agente **executor** é definido pelo par ordenado:

(***Ikb, Cae***) onde:
(i) ***Ikb*** é uma base de conhecimento individual, e
(ii) ***Cae*** é um conjunto de capacidades ≡ {especialidade/perícia, comunicação, aprendizagem}.

Deve-se observar que:

(i) os elementos do par ordenado, ou seja ***Ikb*** e ***Cae*** constituem a competência do agente executor.

Uma estrutura organizacional [fig. 44], é uma disposição de agentes gestores e executores, de forma que cada um coopere com suas competências (conhecimento e capacidades) em duas das dimensões fundamentais da organização:

(i) a dimensão vertical, e

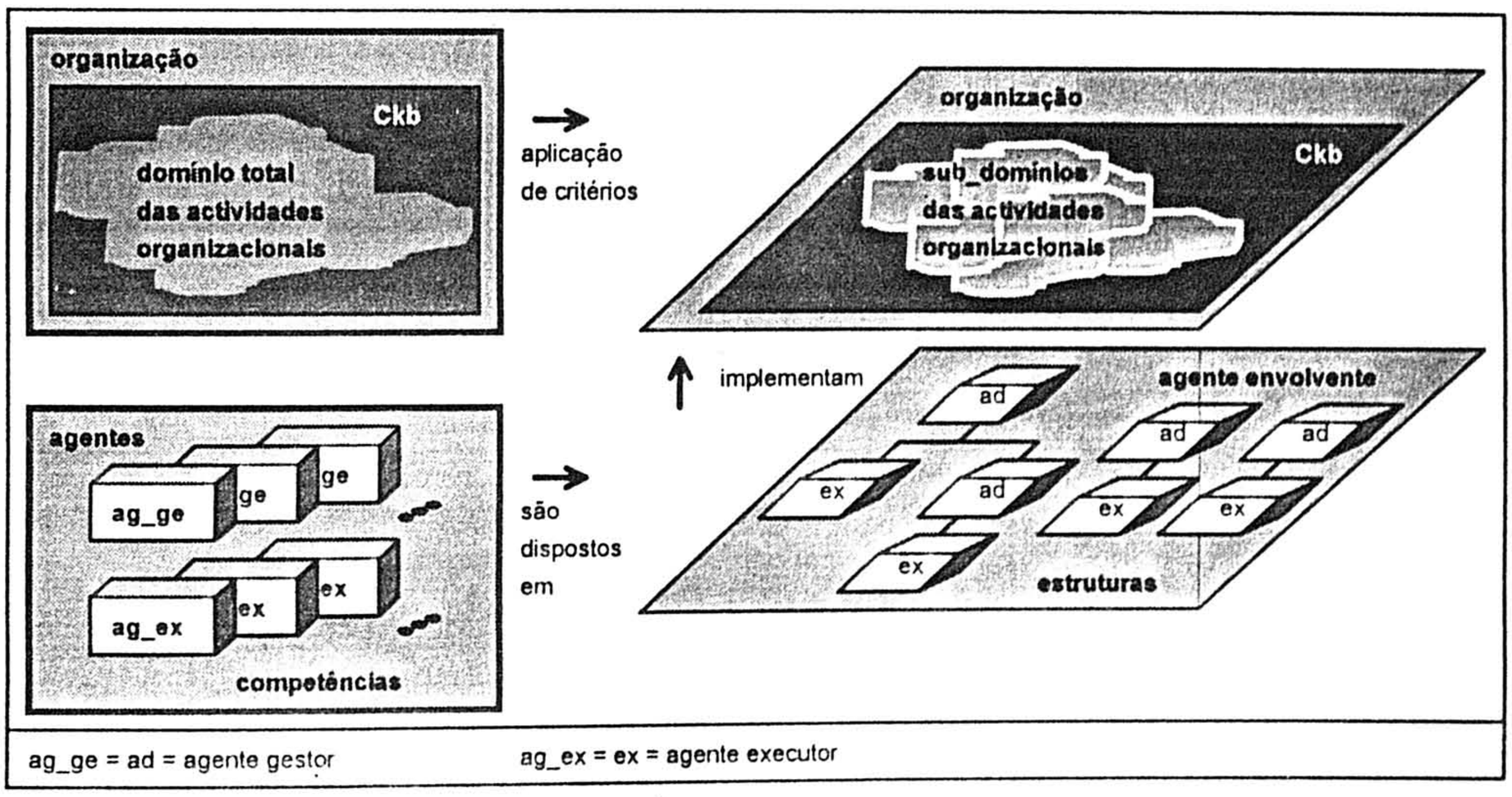
(ii) a dimensão horizontal.

A dimensão vertical diz respeito aos sub-domínios das actividades organizacionais. Esses sub-domínios são produzidos a partir da aplicação de critérios (ver secção 2.2) sobre o total de actividades requeridas pela organização, os quais devem atender as actividades operacionais (tais como produção, vendas, etc.) e as de suporte (tais como finanças, materiais, etc.). A dimensão horizontal diz respeito ao nível de responsabilidade e de decisão por segmentos da organização ou por toda ela. A dimensão horizontal pode ser vista como composta de até três camadas:

(i) uma de mais alta nível (*tl*), a qual é responsável pelo planeamento e coordenação da organização como um todo,

(ii) uma de nível intermediário (*il*) (se existir), a qual é responsável pelo planeamento e coordenação dos sub-domínios, i.e. assegurar a realização das actividades que os compõem, e

(iii) uma de nível mais baixo (*bl*), a qual é responsável pela execução das actividades que compõem os sub-domínios.



[fig. 44] Organização, agentes e estrutura

A disposição dos agentes gestores e executores em um agente envolvente para atender as duas dimensões organizacionais pode variar, de estruturas que enfatizam para mais ou para menos:

(i) a hierarquia (*hi*), a qual ocorre pela:

(i.i) decomposição das actividades que compõem um sub-domínio em tarefas específicas e especializadas, requerendo capacidades também específicas e especializadas dos agentes competentes para tal, e

(i.ii) responsabilidade pelo planeamento e pela realização das actividades que compõem um, a conjunção de alguns ou a conjunção de todos os sub-domínios. Para assegurar a realização dos planos, através do comprometimento das competências necessárias para a sua realização, a autoridade sobre comportamentos específicos (por exemplo: inibir ou activar uma competência, agendar o período de execução, etc.) dos agentes envolvidos na sua execução, é exercida. Esta autoridade, por parte de um agente hierarquicamente superior, ocorre devido a competência e a visão parcial que cada um dos agentes comprometidos com as actividades de um sub-domínio possui, e

(ii) a dinâmica dessas estruturas, a qual se refere a reestruturação para satisfazer as contigências de um mercado e os requerimentos (conceitos e tecnologia) do produto produzido ou serviço prestado.

Uma estrutura com ênfase fortemente hierárquico (*his*), em um agente envolvente, é uma estrutura que dispõe, à partida, todos os agentes gestores e executores comprometidos com a organização, em uma topologia do tipo grafo acíclico, onde:

(i) a raiz representa a camada de mais alto nível (*tl*) da dimensão horizontal, cuja competência para exercê-la é atribuída a um e somente um agente gestor,

(ii) os seus descendentes podem ser folhas ou não. Se forem folhas a competência é atribuída a agentes executores, caso contrário é atribuída a agentes gestores, os quais por sua vez, recursivamente, se enquadram na descrição do item (ii). Os agentes gestores e executores representam respectivamente, a camada de nível intermediário (*il*) e a camada de nível mais baixa (*bl*) da dimensão horizontal, e

(iii) os arcos verticais (dimensão vertical) que unem os agentes representam a hierarquia (*hi*) de uma competência em relação a outra.

Sejam ***Aas*** e ***Eas***.
Uma **organização** dotada de uma estrutura ***his*** é uma tripla tal que $Ohis = (Aas, Eas, Ckb)$, $Aas \neq \emptyset$, $Eas \neq \emptyset$, $Ckb \neq \emptyset$.

Uma estrutura com ênfase fracamente hierárquico (*hiw*), em um agente envolvente, é uma estrutura que dispõe, à partida, todos os agentes gestores e executores comprometidos com a organização em um conjunto (floresta) composto por sub-conjuntos disjuntos de topologias do tipo grafo acíclico, onde:

(i) cada sub-conjunto é composto por:

(i.i) uma raiz, a qual é representada pela competência de um agente gestor,

(i.ii) os seus descendentes são folhas representadas por agentes executores, e

(i.iii) os arcos verticais (dimensão vertical) que unem os agentes (a raiz e seus filhos) representam a hierarquia (*hi*) de uma competência em relação a outra.

(ii) a conjunção das raizes de cada um dos sub-conjuntos representa a camada de mais alto nível (*tl*) da dimensão horizontal, e

(iii) a conjunção das folhas de cada um dos sub-conjuntos, representa a camada de nível mais baixa (*bl*) da dimensão horizontal.

Sejam ***Aas*** e ***Eas***.
Uma **organização** dotada de uma estrutura ***hiw*** é uma tripla, tal que ***Ohiw*** = (***Aas***,***Eas***,***Ckb***), ***Aas*** ≠ ∅, ***Eas*** ≠ ∅, ***Ckb*** ≠ ∅.

Uma estrutura ***his*** é característica de organizações que produzem bens ou serviços com alto nível de padronização das tarefas a serem executadas e cujo mercado se apresenta como de consumo em larga escala. A dinâmica da estrutura é baixa, sendo a reestruturação consequência dos já citados fatores:

(i) contigênciais, e

(ii) de requerimentos dos produtos produzidos.

Por outro lado uma estrutura ***hiw*** é característica de organizações voltadas à produção de produtos personalizados a partir de necessidades e especificações do cliente, com alta especialização das competências necessárias. Este tipo de estrutura apresenta um alto grau de dinamismo, cuja reconfiguração, devido aos projectos, pode torná-las por vezes, porém temporariamente, fortemente hierárquicas.

Entre estes dois tipos fundamentais de estruturas (***his*** e ***hiw***), variações podem ocorrer, enfatizando para mais ou para menos a hierarquia (a presença da segunda camada horizontal) em segmentos de uma mesma organização.

Quanto as metodologias de coordenação, e para ambos os tipos fundamentais de estruturas, a supervisão directa e a padronização são fortemente empregues na dimensão vertical, porém não deixando de haver a de ajuste mútuo com ênfase na argumentação a qual, por vezes, é função do grau de especialização da competência. Para a dimensão horizontal a metodologia empregue é a de ajuste mútuo via negociação e argumentação.

As dimensões, horizontal e vertical e a estrutura adoptada, caracterizam [fig. 45] as actividades dos agentes que compõem um agente envolvente.

dimensões horizontais . nas estruturas ***hiw***	***tl***		***bl***
. nas estruturas ***his*** **características**	***tl***	***ml***	***bl***
. tipo actividade . classe da actividade . iniciatica para o exercício da competência	. planeamento global . não programada . própria	. planeamento dos sub-domínios . programadas e não programadas . própria . ordenada pela hierarquia . padrões . planeamento global	. tarefa do sub-domínio . programadas e não programadas . ordenada pela hierarquia . padrões
. produto da actividade	. objectivos . planos de alto nivel	. plano operacional, das actividades e das competências subordinadas	. geração de produto especifico
. actividade vista na perspectiva da dimensão vertical	. designar as actividades planeadas . coordenar as actividades planeadas . coordenar as competências subordinadas	. designar as tarefas, produto da decomposição das actividades . coordenar as tarefas . coordenar as competências subordinadas	. geração do produto especifico

[fig. 45] Características das actividades nas dimensões e nas estruturas de um agente envolvente

De forma não exaustiva as seguintes características se apresentam:

(i) **quanto ao tipo de actividade**:

(i.i) planeamento global. É a formalização das intenções e desejos que o agente envolvente se propõe realizar e alcançar. O planeamento global orienta os comportamentos quanto as actividades da organização como um todo,

(i.ii) planeamento dos sub-domínios. É a formalização das intenções e desejos que o agente envolvente se propõe realizar e alcançar a nível dos sub-domínios. Este planeamento orienta os comportamento quanto as actividades dos agentes que compõem a estrutura que implementa o sub-domínio, e

(i.iii) tarefa do sub-domínio. É a implementação, através de tarefas, das intenções e desejos formalizados,

(ii) **quanto à classe de actividade**:

(ii.i) programada. São actividades que constituem rotinas i.e. as tarefas que as compõem são padronizadas e periódicas, e

(ii.i) não programadas. São actividades não rotineiras geradas pelo planeamento global ou do sub-domínio. Estas actividades alteram o espaço de estados das actividades do agente envolvente, e podem se tornar programadas,

(iii) **quanto à iniciativa para exercício da competência**:

(iii.i) é própria quando a activação do exercício da competência ocorre devido ao alto grau de autonomia do agente que a exerce e em função da camada horizontal a que está enquadrado,

(iii.ii) é ordenado pela hierarquia quando a activação da competência ocorre devido a autoridade,

(iii.iii) é por padrões quando o agente executa actividades de rotina, e

(iii.iv) é pelo planeamento global quando os planos são utilizados como padrões para activação das competências, característica de estruturas ***hiw***,

(iv) **quanto ao produto da actividade**:

(iv.i) objectivos, os quais são a formalização das intenções e desejos,

(iv.ii) planos de alto nível, divididos em planos estratégicos, tácticos e operacionais, os quais respectivamente estabelecem, em relação ao objectivo: o que fazer para alcança-lo, como fazer para alcança-lo e as actividades a serem decompostas em tarefas ou já executáveis, para realiza-los,

(iv.iii) plano operacional das actividades, o qual contém as tarefas a serem executadas, por parte dos agentes com competência para elas,

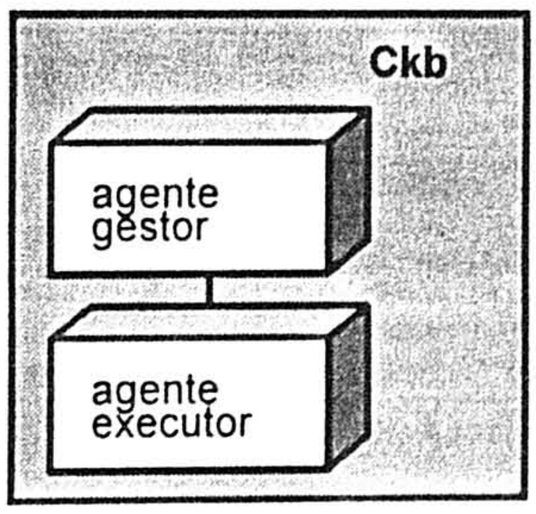
(iv.iv) plano das competências subordinadas, o qual expressa os compromissos atribuídos aos agentes, e

(iv.v) produto específico, o qual é o resultado da execução de uma tarefa, e

(v) **quanto à actividade vista na perspectiva da dimensão vertical**:

a excepção dos agentes executores, cuja competência é a de geração de um produto especificado pela tarefa a eles designada, os agentes gestores tem como actividades as de coordenação (inibir, activar e orientar as competências) das actividades/tarefas.

O agente envolvente de menor granularidade [fig. 46] para modelar uma organização é um agente ***Ohis*** composto de um agente gestor ($a_{agentes} = 1$) e um agente executor ($e_{agentes} = 1$), cujas competências, respectivamente, estão distribuídas nas capacidades de gerir (planear, coordenar, comunicar e aprender) as capacidades de executar (especialidade/perícia, comunicação e aprendizagem).



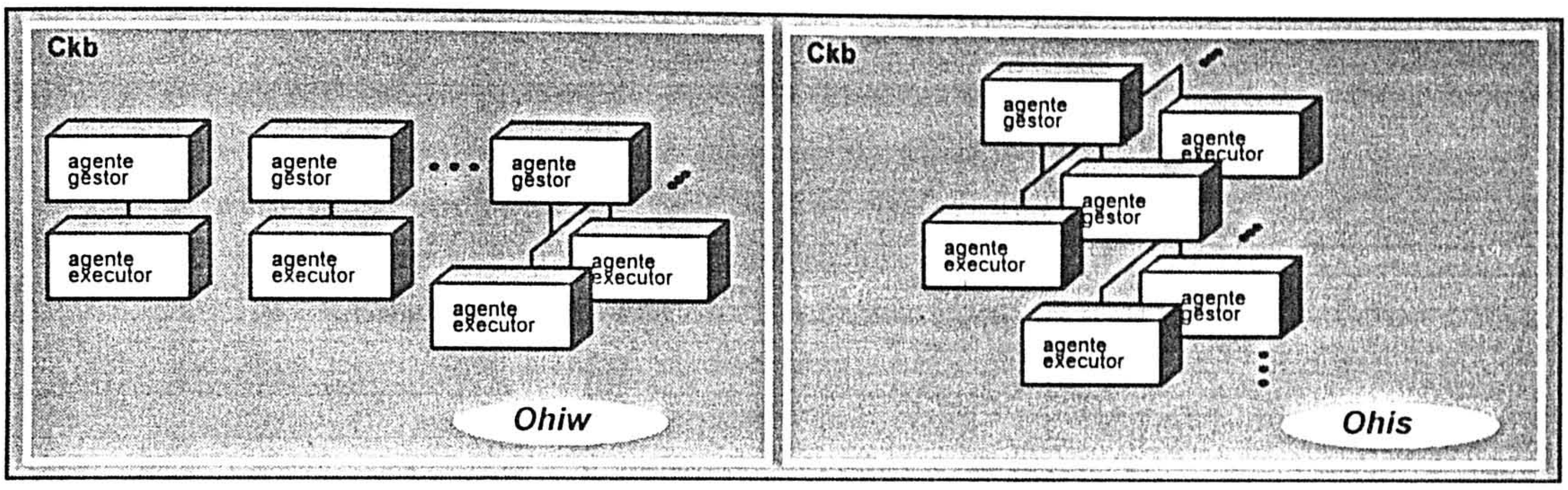
[fig. 46] O agente envolvente de menor granularidade

A distribuição das competências para a elaboração e consecução de objectivos organizacionais em dois tipos de agentes, permite explorar a execução das capacidades de forma modular, estruturada, paralela e com diferentes graus de autonomia. Capacita que diferentes papeis possam ser desempenhados paralelamente pela organização, porém planeados e coordenados de forma integrada aos objectivos

organizacionais, uma vez que as acções executadas pelas competências dos agentes executores são cooperativas. Como exemplo podemos citar o modelamento de uma profissão liberal.

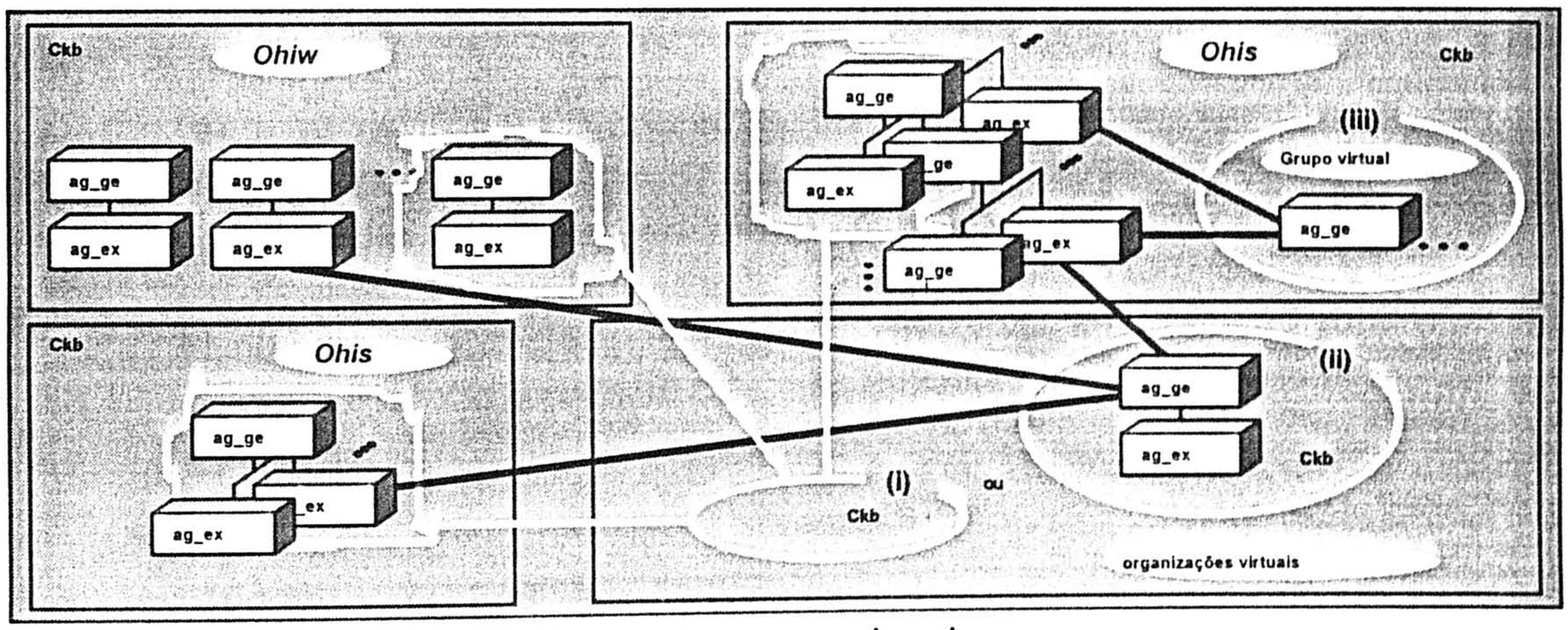
A medida que a complexidade das actividades organizacionais aumenta, a distribuição das competências em competências para a gestão e para a execução em domínios específicos de perícia apresenta vantagens, em relação ao modelamento monolítico (dotar um mesmo agente com competências de gestão e de perícia nos vários domínios), dado que, a gestão das capacidades de execução pode ser estendida e reconfigurada.

A [fig. 47] apresenta as estruturas organizacionais básicas (***Ohiw*** e ***Ohis***), modeladas por agentes envolvente.



[fig. 47] Estruturas organizacionais modeladas através de agentes envolventes

Na formação de estruturas virtuais [fig. 48] utilizando agentes envolventes, o(s) agente(s) participante(s) da estrutura virtual continua(m), do ponto de vista do agente envolvente de origem, a ser(em) parte integrante de seus recursos cognitivos.



[fig. 48] Estruturas virtuais

O facto de continuarem a ser parte integrante dos recursos do agente envolvente de origem permite, a sua alocação a outras tarefas, dentro de suas competências, no período fora de contratação.

Os agentes aquando do desempenho de suas competências numa estrutura virtual estarão, no período contratado (total ou parcialmente), com suas capacidades inibidas na organização de origem e activas na organização ou grupo virtual no qual participam. Se o contratado for um agente executor, este passa a ter o seu comportamento orientado pelo gestor ao qual está vinculado no período. Se o

contratado for um gestor com algumas ou com todas as competências a ele vinculadas, a forma de participação numa estrutura virtual com ênfase ***his*** ou com ênfase ***hiw***, é determinante para a aplicação das metodologias de coordenação (supervisão directa e padronização de procedimentos no caso da ênfase fortemente hierárquica, negociação ou argumentação no caso da ênfase fracamente hierárquica).

Uma estrutura virtual é uma estrutura que nasce a partir de outras estruturas já existentes, sendo a sua criação resultado de interacções sociais (compromisso baseado em contracto limitado no tempo) com os demais elementos activos de uma sociedade.

Ainda com referência a [fig. 48] são dados três exemplos de agentes envolventes: dois ***Ohis*** e um ***Ohiw***, dos quais são contratadas algumas competências para formação das seguintes estruturas virtuais:

(i) uma organização virtual totalmente composta por agentes gestores e executores de diferentes agentes envolventes. A estrutura virtual criada enfatiza fracamente a hierarquia, onde a coordenação ocorre através da negociação e argumentação, acentuando a autonomia dos agentes quanto a aplicação de suas competências.

A dimensão horizontal ***tl*** é representada pela conjunção dos agentes gestores e a ***bl*** é representada pela conjunção dos agentes executores, ambas pertencentes aos três agentes envolventes participantes.

Sejam Aas e Eas.
Uma **organização** dotada de uma estrutura virtual com ênfase ***hiw*** é uma tripla, tal que ***Vhiw*** = (***Aas,Eas,Ckb***), ***Aas*** $\neq \emptyset$, ***Eas*** $\neq \emptyset$, ***Ckb*** $\neq \emptyset$.

(ii) a partir de uma organização ***Ohis*** já constituída, a qual é composta por um agente gestor e um executor, tendo como virtual as competências de agentes executores contratados de outros agentes envolventes. A estrutura virtual criada enfatiza fortemente a hierarquia, acentuando a coordenação através de metodologias de supervisão directa e padronização, não excluindo, contudo, o uso da argumentação,

Sejam ***Aas*** e ***Eas***.
Seja ***Edas*** o conjunto de agente executores $\in$ {$Ag_1, Ag_2, \ldots, Ag_n$}.
Uma **organização** dotada de uma estrutura virtual com ênfase ***his*** é uma quádrupla, tal que ***Vhis*** = (***Aas,Eas,Edas,Ckb***), ***Aas*** $\neq \emptyset$, ***Eas*** $\neq \emptyset$, ***Edas*** $\neq \emptyset$, ***Ckb*** $\neq \emptyset$.

e,

(iii) o terceiro caso apresentado, ainda com referência a [fig. 48], é a formação de um grupo virtual. As competências de uma mesma organização, as quais estão vinculadas a diferentes gestores, encontram-se reunidas sob a gestão específica de um (podendo ou não, ser um dos gestores de origem de uma das competências). Durante o período acordado (total ou parcial), os agentes executores estarão sujeitos a orientação do agente gestor a que passam a estar vinculados. As metodologias de coordenação empregues são as mesmas referidas no item (ii).

Sejam ***Aas*** e ***Eas***.
Um **grupo virtual** com ênfase ***his*** de uma organização ***Ohis*** é um par ordenado, tal que ***Gvhis*** = (***Aas***,***Eas***), ***Aas*** ≠ ∅, ***Eas*** ≠ ∅.

O conhecimento corporativo

Existem dois diferentes conjuntos de conhecimentos [94] presentes em um agente envolvente:

(i) o conhecimento corporativo (*Ck*) [fig. 49], e

(ii) o conhecimento individual (*Ik*),

os quais respectivamente constituem a *Ckb* (base de conhecimento corporativa) e a *Ikb* (base de conhecimento individual).

O *Ck* e o *Ik* são adquiridos, transformados e aprendidos pelos agentes gestores e executores durante as interacções intra e inter agentes envolventes.

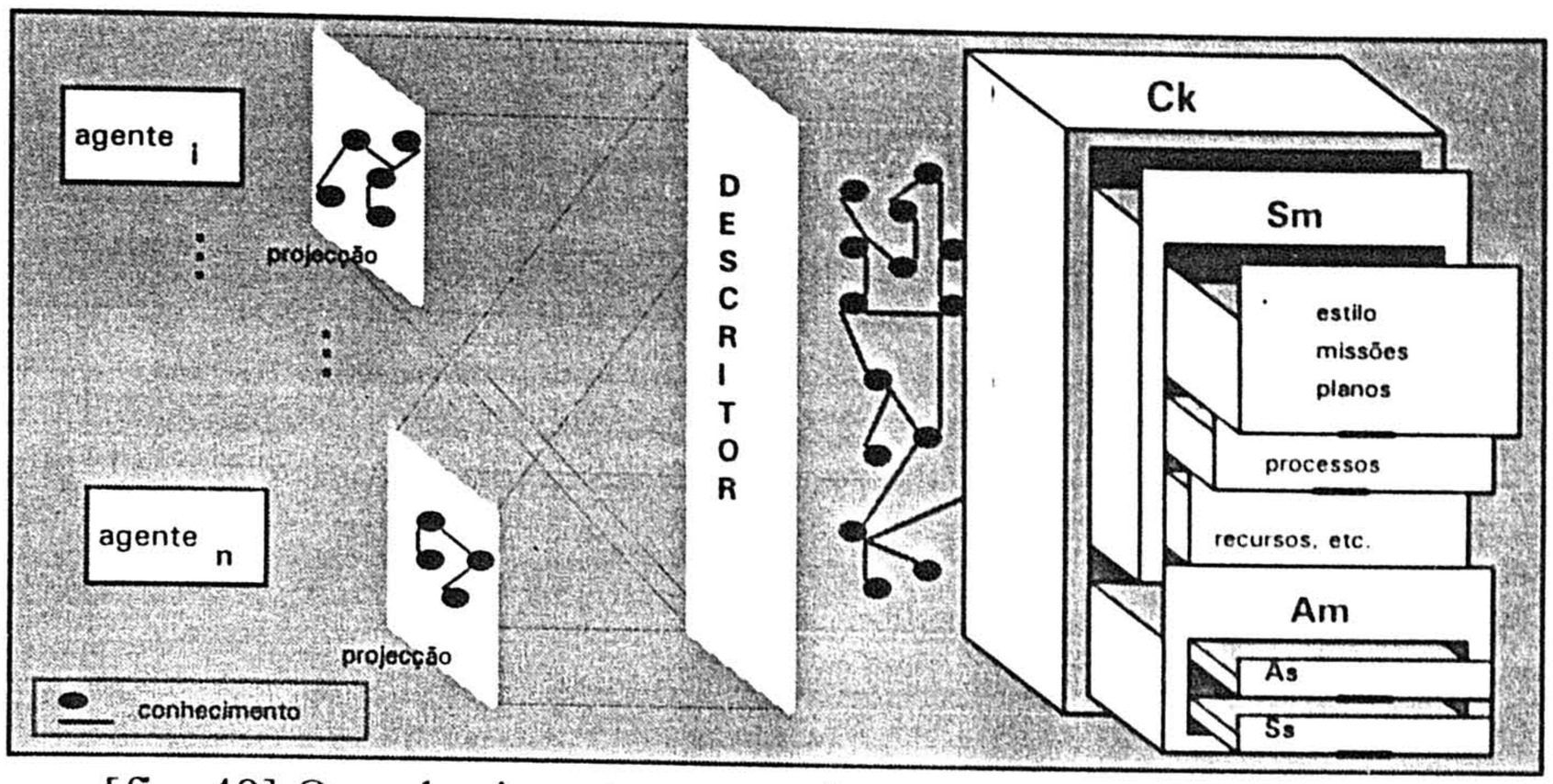
Conceptualmente o conhecimento corporativo (*Ck*) é o conhecimento que caracteriza o agente envolvente como uma organização, e cujo conteúdo não pertence, em particular, a nenhum de seus componentes (agentes gestores e executores), porém é utilizado como um dos componentes orientador de comportamentos. O *Ck* é constituído por dois modelos [63]:

(i) o modelo do auto conhecimento (*Sm*), que representa o ambiente interno de um agente envolvente, e

(ii) o modelo dos agentes conhecidos (*Am*), representa o ambiente externo ao agente envolvente.

O modelo dos agentes conhecidos (*Am*) é dividido em dois sub-modelos: o modelo dos demais agentes envolventes (*As*), organizações) e os indivíduos, que constituem uma sociedade, e o modelo da sociedade (*Ss*), ambos contendo de forma dinâmica a representação local do conhecimento obtido e aprendido durante as interacções.

O conhecimento corporativo (*Ck*) é uma base de conhecimento partilhada, a qual é acessível e manutenível pelos agentes gestores e executores que compõem o agente envolvente. A acessibilidade e a manutenabilidade são baseadas em um modelo de permissões (descritor) que habilita os agentes internos a organização terem uma visão compatível (projecção) com suas competências, e actividades pelas quais são responsáveis.



[fig. 49] O conhecimento corporativo em um agente envolvente

O conhecimento individual (***Ik***) é uma base de conhecimento mantida especificamente por cada um dos agentes gestores e executores a que esta pertence, porém é acessível sob pedido, devido ao conhecimento nela contido ser utilizável em actividades executadas por outros agentes. Esta base contém o conhecimento para o exercício da competência (***Ca***) por parte do agente e sobre o sub-domínio (***Sd***) com o qual o mesmo está envolvido, em função da actividade organizacional que desempenha.

2.5.3.2 Perspectiva Computacional para um Agente Envolvente

Quando uma estação de trabalho, recebe a mensagem para executar o ficheiro "*ARTOR_start_society*" (citado na secção 2.5), uma cópia do "*YAP*"[1] passa a ser carregada para cada um dos componentes da sociedade que foram nomeados pelo **descritor_do_artefacto_"*artor***". Os agentes envolventes ("*cover_agents*"), componentes da sociedade, são definidos pelo par ordenado:

(***Kb***, ***Asca***) onde:
(i) ***Kb*** é a base de conhecimento, e
(ii) ***Asca*** é um modelo, de mecanismo genérico, para a geração automática de agentes envolventes,

cujo mecanismo genérico ("*artor_standard_cover_agent*"), que gera cada um dos agentes envolventes nomeados no descritor_do_artefacto_"*artor*", contém os seguinte predicados "*YAP*":

```
agent:-
```

[1] Dada a natureza declarativa do "Prolog", ...zes ...liza do progr... na explicação descritivas dos agentes.

```
        % identificação do agente envolvente, conforme descritor_do_artefacto_"artor"
i_am( Agent),
        % obtenção da identificação da estação de trabalho
host_name( Host_name),
        % procedimento de carga da base de conhecimento corporativa (Kb)
loading_corporate_knowledge( Agent, Host_name),
        % procedimento de criação do suporte computacional para as comunicações
creating_communication_support( Agent, Host_name),
        % com base na base de conhecimento corporativa executa o lançamento

        % dos agentes que constituem o agente envolvente
competence_to_trigger( Agent, Host_name),
        % ciclo que constitui a existência do agente envolvente
agent_life( Agent, Host_name).
```

O procedimento de criação do suporte computacional para as comunicações, representado pela cláusula *creating_communication_support(Agent, Host_name),* executa a primitiva de geração de "*sockets*", disponível na versão do "*YAP*":

```
create_socket( +Host_name, +PortNumber, -SocketDescriptor).
```

O argumento "*PortNumber*" é obtido do descritor_do_artefacto_"*artor*", e o argumento "*SocketDescriptor*", o qual passa a ser o elemento de identificação interna do agente para a comunicação, é retornado pela execução do predicado.

As cláusulas "*loading_corporate_knowledge*" e "*competence_to_trigger*" executam o lançamento de uma cópia do "*YAP*", respectivamente, para a base de conhecimento corporativa (***Kb***) e tantas quantas forem os agentes que compõem cada um dos agentes envolventes.

```
/*****************************************************************/
competence_to_trigger( Agent, Host_name):-
        % competência para disparar os componentes contidos na Kb
  is_competent_start( Agent, Component_type),
        % instância para o tipo de componente
  is_a( Component_instance, Component_type),
        % procedimento que lança uma versão do "YAP" com o mecanismo
        % genérico "artor_standard_adm_exe_agent" [2] de
        % geração de agentes gestores e executores.
  starting_components( Component_instance, Component_type),
  fail.

competence_to_trigger( _, _):-
  !.
/*****************************************************************/
```

O exemplo a seguir, apresenta um segmento da base de conhecimento corporativa ("***Ckb***") de um agente envolvente denominado de "*artor_org_A*".

```
/*****************************************************************/
i_am( artor_org_A_corporate_knowledge_data_base).
/*****************************************************************/
% Self Model - modelo do auto conhecimento
/*****************************************************************/
% Self Model - Sobre os recursos organizacionais
```

[2] O mecanismo genérico "*artor_standard_adm_exe_agent*" é adiante descrito

```
/*****************************************************/
is_a( artor_org_A, cover_agent).
%
is_a(artor_org_A_corporate_knowledge_data_base,
  corporate_knowledge_data_base).
is_a( artor_org_A_director, administrator_agent).
is_a( artor_org_A_comercial_manager, administrator_agent).
is_a( artor_org_A_selection_expert, executor_agent).
%
is_part_of( artor_org_A_corporate_knowledge_data_base,
  artor_org_A).
is_part_of( artor_org_A_director, artor_org_A).
is_part_of( artor_org_A_comercial_manager,
  artor_org_A_director).
is_part_of( artor_org_A_selection_expert,
  artor_org_A_comercial_manager).
%
is_competent_start( artor_org_A, administrator_agent).
is_competent_start( artor_org_A, executor_agent).
is_competent_start(artor_org_A,
corporate_knowledge_data_base).
/*****************************************************/
```

O ciclo que consubstancia a existência do agente envolvente é fornecido pelos seguintes predicados:

```
/*****************************************************/
%                  *********  agent_life *********
/*****************************************************/
agent_life( Agent, Host_name):-
  repeat,
          % leitura das mensagens e distribuição das mesmas aos componentes
          % do agente envolvente
  public_relations,
  fail.
/*****************************************************/
```

O agente envolvente executa duas funções básicas:

(i) lançar os componentes que o constituem, i.e. a base de dados corporativa e com base nela os agentes gestores e executores, e

(iii) encerrar a actividade dos componentes e por consequência desactivar a organização. O pedido de encerramento é solicitado através do "*artor_human_being_machine_interface_status*".

Os agentes gestores e executores, são definidos pelo par ordenado:

(***Kb, Asaea***) onde:

(i) ***Kb*** é a base de conhecimento, e

(ii) ***Asaea*** é um modelo, de mecanismo genérico, para a geração automática de agentes gestores e executores,

cujo mecanismo genérico "*artor_standard_adm_exe_agent*", que gera cada um dos agentes lançados pelo agente envolvente, contém os seguintes predicados "*Yap*"

```
agent:-
        % identificação do agente, conforme lançado pelo agente envolvente
  i_am( Agent),
        % obtenção da identificação da estação de trabalho
  host_name( Host_name),
        % procedimento de carga das capacidades do agente gestor/executor
  my_capacities( Agent, Host_name),
        % ciclo que constitui a existência do agente gestor/executor
  agent_life( Agent, Host_name).
```

O procedimento de carga das capacidades "*my_capacities*" para os agentes gestores/executores, consulta a base de conhecimento corporativa para obter quais são as capacidades do agente, e quais os módulos que as implementam. Este procedimento possibilita que, de forma dinâmica, seja possível carregar a base de conhecimento individual e as diferentes de capacidades do agente através dos respectivos módulos que as implementam.

```
/*************************************************************/
my_capacities( Agent, Host_name):-
        % Identifica a capacidade e o módulo que a implementa.
        % Estes conhecimentos são obtidos da base de conhecimento corporativa
  is_a( Module, Capacity, Agent, Capacity_sequence),
        % Localização do módulo, no descritor_dos_artefactos,
        % o qual implementa a capacidade
  is_located( artifact, artor_module, Module, Host_name,
              Addr),
        % procedimento para a carga do módulo que implementa uma capacidade do agente
  set_up_addressing( Module, Host_name, Addr, Addr_ready),
  fail.

my_capacities( _, _):-
  !.
/*************************************************************/
```

O predicado "*is_a/4*" contém um dos seus argumentos definido como "*Capacity_sequence*". Este argumento o qual pode ser alterado em tempo de existência do agente, contém:

(i) um ordinal, o qual explicita a prioridade na qual as capacidades são executadas durante o ciclo de vida do agente, quando o mesmo não tem o fluxo do exercício de suas capacidades orientado:

(i.i) por uma mudança de estado do seu universo observável, ou

(i.ii) pelo recebimento de uma mensagem, ou

(ii) o valor "*started_by*" quando sua capacidade é exercida quando da recepção de uma mensagem solicitando sua activação.

Como exemplo, os predicados que seguem, fazem parte da base de conhecimento corporativa de um agente envolvente chamado de "*artor_org_A*", referentes ao agente gestor "*artor_org_A_director*" e ao agente executor "*artor_org_A_selection_expert*".

```
/*************************************************************/
```

```
i_am( artor_org_A_corporate_knowledge_data_base).
/*************************************************/
% Self Model - modelo do auto conhecimento
/*************************************************/
% Self Model - Sobre os recursos organizacionais
/*************************************************/

is_responsible_for( artor_org_A_director,
  conduct_organization).
is_responsible_for( artor_org_A_selection_expert,
  selecting_goods).
%
is_part_of( coordination_capacity, artor_org_A_director).
is_part_of( planning_capacity, artor_org_A_director).
is_part_of( communication_capacity, artor_org_A_director).
is_part_of( learning_capacity, artor_org_A_director).
%
is_a( artor_coor_c1, coordination_capacity,
  artor_org_A_director, 3).          % The coordination_module
is_a( artor_plan_px_catch_all, planning_capacity,
  artor_org_A_director, 2).          % The planning_module
is_a( artor_comm_m1, communication_capacity,
  artor_org_A_director, 1).          % The communication_module
is_a( artor_lear_12, learning_capacity, artor_org_A_director,
4). % The learning_module
%
is_part_of(                                  selection_capacity,
artor_org_A_selection_expert).
is_part_of( communication_capacity,
        artor_org_A_selection_expert).
is_part_of( learning_capacity, artor_org_A_selection_expert).
%
is_a( artor_sele_s1, selection_capacity,
  artor_org_A_selection_expert,       started_by).%        The
expert_module
is_a( artor_comm_m1, communication_capacity,
  artor_org_A_selection_expert,1).                 %        The
communication_module
is_a( artor_lear_11, learning_capacity,
  artor_org_A_selection_expert,started_by).%The
learning_module
%
has_environmental_att( artor_org_A_director, turing,
  3610, [2, 168, 400, 100, green]).
has_environmental_att( artor_org_A_selection_expert, turing,
  3620, [842, 2, 400, 200, green]).
/*************************************************/
```

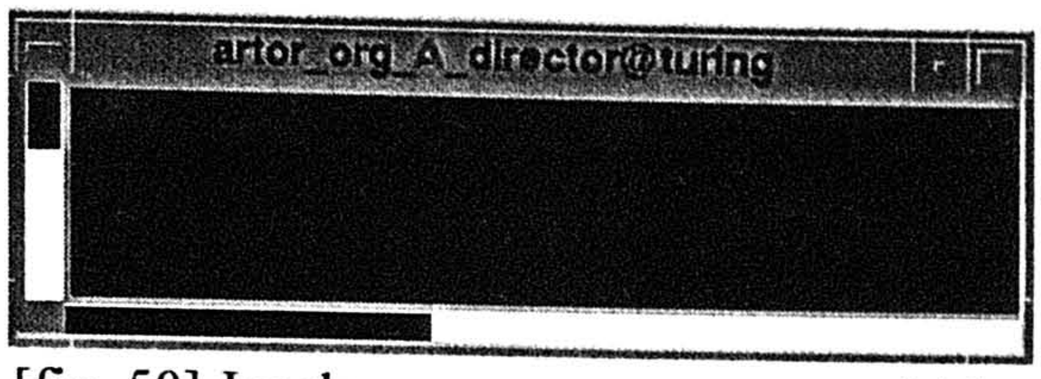
O predicado "*has_environmental_att*", especifica o ambiente de tecnologia e de apresentação no qual a sociedade será lançada:

(i) o conhecimento relativo a estação de trabalho (exemplo "*turing*") na qual o agente será lançado,

(ii) o porto de identificação do agente para efeito de comunicação (exemplo *3610*), e

(iii) uma lista contendo característica para efeito de interface (*[842, 2, 400, 200, green]*).

Para cada agente lançado, uma janela [fig. 50] é criada com base no predicado "*has_environmental_att*". Esta janela apresentará as mensagens recebidas durante a comunicação intra e inter-agentes envolventes.



[fig. 50] Janela para as mensagens recebidas

A localização dos módulos que implementam as capacidades estão contidos no descritor_dos_artefactos.

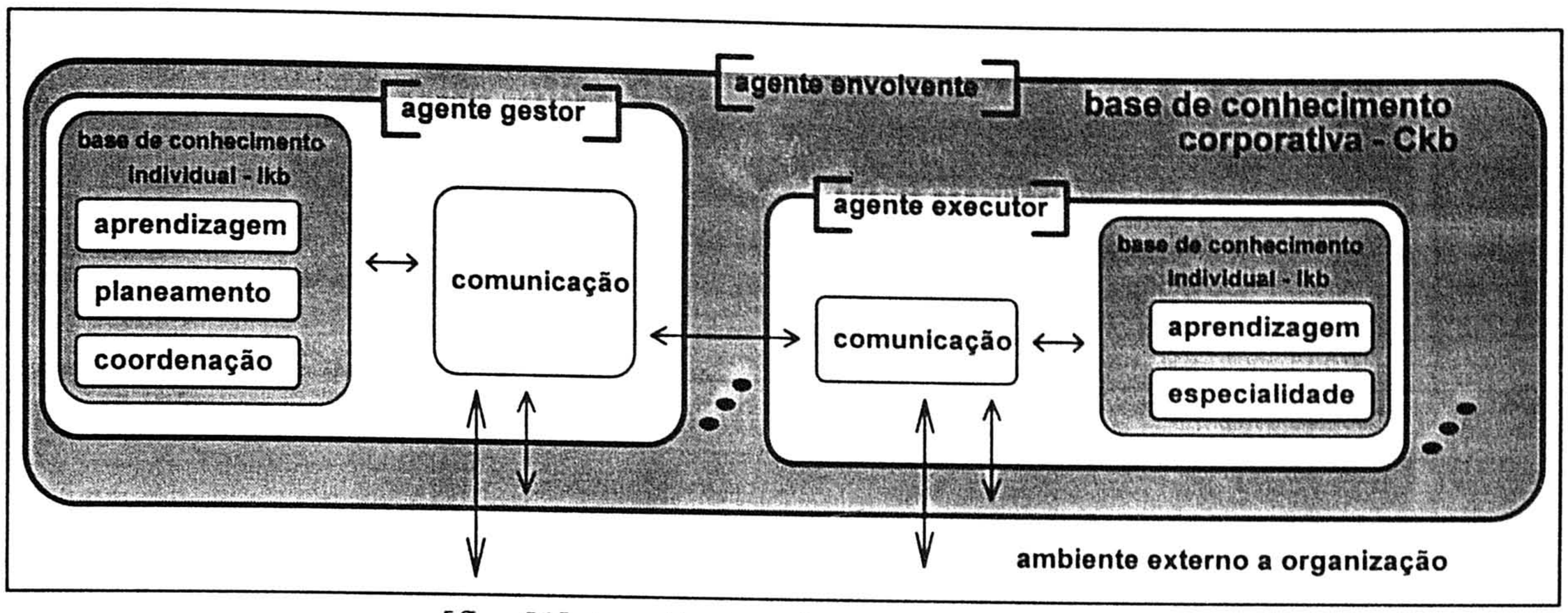
```
% i_am(artifact artor_artifact_descriptor).
%
is_located(artifact artor_module, artor_coor_c1, turing,
  ['/usr/users/shmeil/ARTOR.dir']).
...
```

O ciclo que consubstancia a existência dos agentes gestores e executores é fornecido pelos seguintes predicados "*YAP*":

```
/*****************************************************/
%              *********  agent_life  *********
/*****************************************************/
agent_life( Agent, Host_name):-
  repeat,
  capacities( Agent),
  fail.
/*****************************************************/
capacities( Agent):-
  findall((Sequence, Capacity),
        is_a(      Module_name,      Capacity,      Agent,
Sequence),Steps),
  sort(Steps, Sort_steps),
  selecting_next_step( Sort_steps).
/*****************************************************/
selecting_next_step( []):-
  !.

selecting_next_step([(Sequence, Capacity)|Sort_steps_tail]):-
  (
  ( Sequence\==started_by->
    Capacity,
    selecting_next_step( Sort_steps_tail)
  );
  ( selecting_next_step( Sort_steps_tail)
  ).
/*****************************************************/
```

Uma vez activada a comunidade de agentes, interna ao agente envolvente, a definição conceptual organizacional, descrita no início desta secção, passa a ser atendida pela arquitectura apresentada na [fig. 51]:



[fig. 51] Arquitectura dos agentes

(i) uma **organização** é definida por:

Sejam ***Aas*** e ***Eas***.
Uma **organização** é uma tripla, tal que ***Org*** = (***Aas***, ***Eas***, ***Ckb***), ***Aas*** $\neq \emptyset$, ***Eas*** $\neq \emptyset$, ***Ckb*** $\neq \emptyset$.

(ii) um agente **gestor** é definido pelo par ordenado:

(***Ikb***, ***Caa***) onde:
(i) ***Ikb*** é uma base de conhecimento individual, e
(ii) ***Caa*** é um conjunto de capacidades $\equiv$ {planeamento, coordenação, comunicação, aprendizagem}.

Deve-se observar que:
(i) os elementos do par ordenado, ou seja ***Ikb*** e ***Caa*** constituem a competência do agente gestor.

(iii) um agente **executor** é definido pelo par ordenado:

(***Ikb***, ***Cae***) onde:
(i) ***Ikb*** é uma base de conhecimento individual, e
(ii) ***Cae*** é um conjunto de capacidades $\equiv$ {especialidade/perícia, comunicação, aprendizagem}.

Deve-se observar que:
(i) os elementos do par ordenado, ou seja ***Ikb*** e ***Cae*** constituem a competência do agente executor.

Quanto aos módulos que implementam as capacidades de cada um dos agentes, os mesmos dependem da metodologia a qual o utilizador deseja implementar. A descrição que segue, apresenta uma versão experimental dos módulos que implementam as capacidades de coordenação e comunicação dos agentes, deixando os

módulo de aprendizagem e de especialidade de agentes executores para serem contemplados no Capítulo 3. Os exemplos que serão apresentados para os módulos descritos na continuidade desta secção, são actividades organizacionais desempenhadas por agentes gestores, na coordenação de actividades de competência das camadas alta e intermédia da dimensão horizontal. Estas actividades representam o exercício das competências gerenciais as quais antecedem a realização, por parte dos agentes executores, da relação social de selecção e contratação de colaboradores, tema do Capítulo 3 desta dissertação.

2.5.3.2.1 A comunicação

As comunicações intra e inter agentes envolventes, são divididas em dois níveis:

(i) o nível básico, o qual suporta o envio ("*write*") de mensagens e a possibilidade de aceder as recebidas ("*read*"). Este nível é implementado através de "*sockets*" suportados pelos seguintes predicados "*YAP*":

```
write_socket( +SocketDescriptor,
              +HostName,
              +PortNumber,
              +AgentName,
              +Msg ).

read_socket( +SocketDescriptor,
             -AgentName,
             -Msg ).
```

onde:

"+" significa que o argumento deve ser instanciado previamente à execução do predicado,
e
"-" significa que o valor do argumento é retornado como produto da execução do predicado.

"***SocketDescriptor***" é o identificador do socket, para utilização interna do agente. Este identificador é obtido a partir da execução do predicado "*create_sockets*". Refere-se ao agente que está executando o predicado,
"***HostName***" é o endereço da estação de trabalho do agente ao qual a mensagem se destina,
"***PortNumber***" é a identificação do porto do agente ao qual se destina a mensagem,
"***AgentName***" é a identificação do agente de destino, e
"***Msg***" é o conteúdo da mensagem.

Os argumentos "***HostName***" e "***PortNumber***" são valores obtidos das seguintes fontes:
(i) quando são componentes da sociedade, e não houve comunicação entre eles, estes argumentos são obtidos no descritor_do_artefacto_"*actor*",
(ii) repassados via a mensagem, e
(iii) quando são componentes de um agente envolvente, estão contidos na base de conhecimento corporativa.

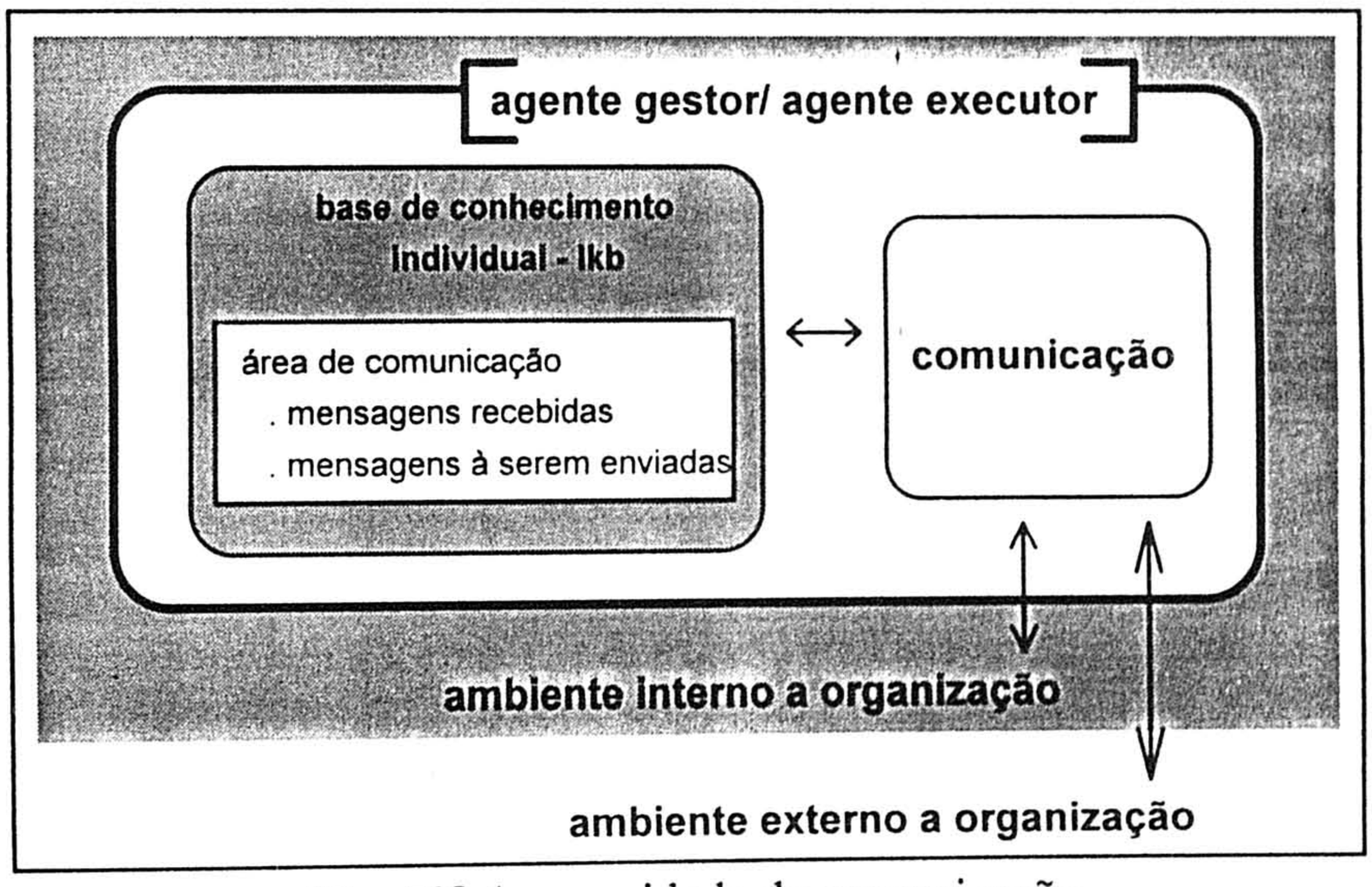
(ii) o nível que representa os canais formais (mensagens formalizadas e reguladoras da autoridade, da actividade e dos recursos) e informais de uma organização.

A estrutura geral da mensagem (nos predicados que suportam os "*sockets*" é o argumento "***Msg***") é uma lista:

```
"Msg" = [message_type, message_content]

onde:
"message_type" = {outer_message, inner_message}
  "outer_message" é indicativo de uma mensagem inter agentes
        envolventes
  "inner_messsage" é indicativo de uma mensagem intra agente
        envolvente
"message_content"     =     (The_message_itself,     Hostname,
PortNumber)
  "The_message_itself" é o conteúdo da mensagem,
  "Hostname" é a identificação da estação de trabalho do
agente          emissor da mensagem,
  "PortNumber" é a identificação do porto do agente emissor
da              mensagem.
```

A versão que implementa a capacidade de comunicação nos agentes, é a contida no módulo "*artor_comm_m1*" [fig. 52], referida na base de dados corporativa. Sua composição fundamental é dada pelos seguintes predicados "*Prolog*" ("*YAP*"):



[fig. 52] A capacidade de comunicação

```
/*****************************************************/
% Communication capacity  module: artor_comm_m1
/*****************************************************/
communication_capacity:-
  i_am( Agent),
  host_name( Host_name),
  creating_communication_support( Agent, Host_name,
                SocketDescriptor),
  public_relations( SocketDescriptor),
  !.
/*****************************************************/
```

O procedimento "*creating_communication_support*" cria o suporte de comunicação para o agente, a exemplo do procedimento utilizado pelo agente envolvente.

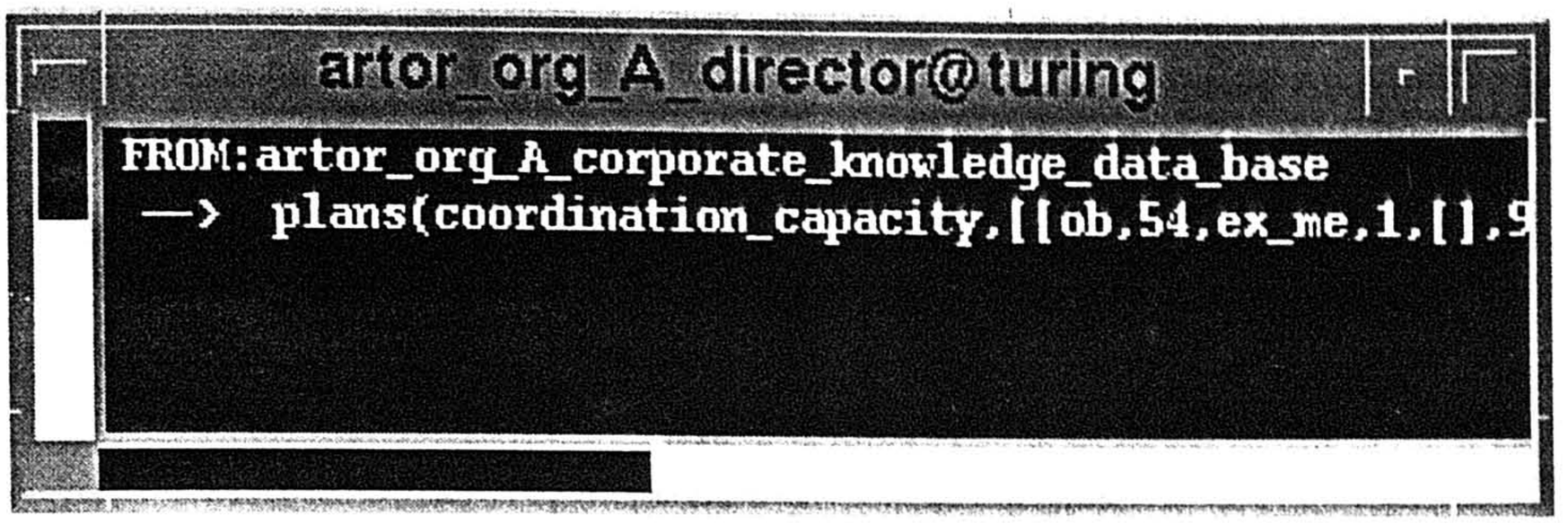
O procedimento "*public_relations*" executa três funções:

(i) a leitura de mensagens oriundas dos demais agentes da comunidade, disponibilizando-as na base de conhecimento individual do agente. O local de disponibilização das mensagens na base de conhecimento individual é denominado de área de comunicação. Esta área de comunicação é uma base de dados interna ao "*YAP*", a qual é mantida através de predicados [92] de leitura e gravação. Sua estrutura é:

```
/*****************************************************/
% The Individual Knowledge - IK
/*****************************************************/
% Internal communication area for all capacities
% The records in this area have the following format:
/*****************************************************/
%
%:-recorda( message_received,
%        message_received( From_agent,
%                Message_type,
%                Message,
%                Host_name_from_agent,
%                PortNumber_from_agent),
%        Message_received_rec_number).
%
%:-recorda( message_to_send,
%        message_to_send( To_agent,
%                Message_type,
%                Message,
%                Host_name_to_agent,
%                PortNumber_to_agent),
%        Message_to_send_rec_number).
%
/*****************************************************/
```

Os argumentos (*Message_received_rec_number* e *Message_to_send_rec_number*) identificam univocamente as entradas na base de dados e são providenciados pelo próprio predicado.

(ii) a apresentação na janela [fig. 53], das mensagens recebidas, a qual foi criada para cada um dos agentes internos ao agente envolvente, e



[fig. 53] Janela para as mensagens recebidas

(iii) o envio das mensagens, aos agentes destinatários, que foram geradas pelas capacidades e por elas dispostas, na base de conhecimento individual, na área de comunicação.

2.5.3.2.2 A base de conhecimento corporativa

Para além do conhecimento sobre os agentes que compõem um agente envolvente, e não na sua totalidade, a base de conhecimento corporativa regista também o conhecimento sobre o perfil do agente envolvente e seus planos. Estes conhecimentos, produto do exercício da competência de planeamento da camada de mais alto nível da dimensão horizontal e gerados com ou sem participação dos agentes que representam as demais camadas, dependendo do tipo de gestor ou gestão (centralizadora, participativa, etc.), orientarão o comportamento dos agentes internos no desempenho de suas competências.

O conhecimento sobre o perfil do agente envolvente faz parte do modelo do auto conhecimento do agente e é representado pelo predicado:

```
skill( Type, Agent, Type_ Value).

onde:    Type é o tipo de perfil do agente,
         Agent é a identificação do agente, e
         Type_value é a instância referente ao tipo.

/*****************************************************/
i_am( artor_org_A_corporate_knowledge_data_base).
/*****************************************************/
% Self Model - modelo do auto conhecimento
/*****************************************************/
% Self Model - Sobre o perfil do agente
/*****************************************************/
skill( mission, artor_org_A, develop_local_industry).
skill( style, artor_org_A, win_win).
...
```

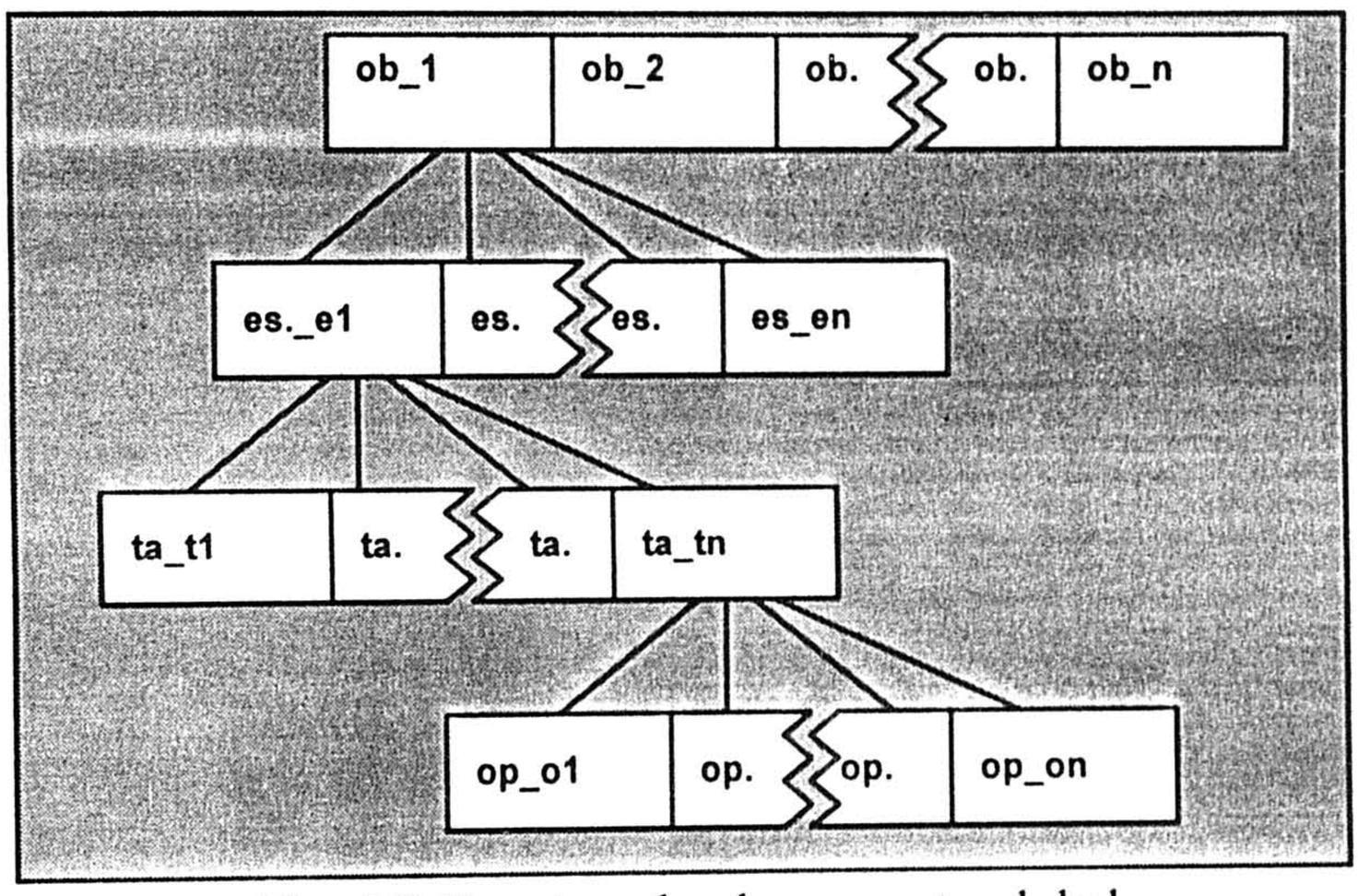
No processo de tomada de decisão, o conhecimento acerca do perfil do agente, actua como orientador do mesmo. Por exemplo, em um processo de negociação:

(i) a preferência será acentuada caso o agente envolvente, potencial parceiro, seja uma industria local, e

(ii) o estilo "*win_win*", considerará as contra-ofertas do potencial parceiro, relaxando alguns dos critérios utilizados para negociar.

Os planos globais de um agente envolvente, expressa uma alteração no espaço de estados interno do agente. Este tipo de plano, normalmente de "*per se*" inclui actividades a serem executadas não tidas como rotineiras na organização (poderão vir a ser), as quais são produto de observações activas e passivas do(s) agente(s) gestore(s).

A representação dos planos na base de conhecimento corporativa é dada pela seguinte estrutura [fig. 54]:



[fig. 54] Estrutura do planeamento global

Os planos são definidos por uma lista de "***n***" objectivos, onde cada objectivo contém uma lista de "***n***" estratégias e assim sucessivamente até o nível operacional.

```
nível_objectivo [... ,[ob_n', id, de, pr, pc, dl, st, cl,
nível_estratégico                [... ,[es_en',
nível_táctico                           [... ,[ta_tn',
nível_operacional                              [... ,[op_on]
                                               ]
                                        ]
                                ]
                        ].
```

A estrutura de cada uma das ocorrências da lista é definida por:

(i) tipo do nível da ocorrência = {**ob**, **es**, **ta**, **op**}
(ii) **id** =identificador da ocorrência,
(iii) **de** = descrição da ocorrência,
(iv) **pr** = prioridade = {1,...,n). Caso exista, substitui outros critérios para escolha da ocorrência,

(v) **pc** = lista de pré-condições que devem estar satisfeitas para que a ocorrência possa ser realizada
[pre_condicao_1(...), ..., pre_condicao_n(...)],
Ex: pré-condição:
pre_condicao = tempo(Operador, Data_desejada)
Operador = { >,<, =} onde > é maior ou igual,
(vi) **dl** = data limite para a realização da ocorrência,
(vii) **st** = estado da ocorrência = {w, i, a, n} onde:
w = ocorrência em espera,
i = ocorrência em realização,
a = ocorrência realizada,
n = ocorrência não realizada,
(viii) **cl** = enquadramento do assunto, o qual identifica a ocorrência em um sub_domínio de actuação da organização.

Exemplo, parcial, de um plano global do agente envolvente "*artor_org_A*":

```
/*****************************************************/
i_am( artor_org_A_corporate_knowledge_data_base).
/*****************************************************/
% Self Model - modelo do auto conhecimento
/*****************************************************/
% Self Model - Sobre os planos
/*****************************************************/
objectivos_planos_agente( [
[ ob, 54, expandir_mercado, 1, [], 970401, w,
  conduct_organization,
        [
[       es,       5401,       lançar_novo_produto,
1,[tempo(>,970202)],
       970320, w, conduct_organization,
       [

       [ ta, 540101, contratar_consultadoria, 1,
             [tempo(>,970205),
             recursos_financeiros(200)],
             970313, w, commercial_subjects,
             [

             [            op,              5401011,
       seleccionar_fornecedores,
                   1,[],970305,                    w,
                   selecting_goods,[]]
             ]
                   ]
                   ]
             ]
       ]
                   ]).
```

Neste exemplo, o objectivo do agente envolvente "*artor_org_A*" é expandir o seu mercado de actuação. Para o alcance do objectivo, sua estratégia é lançar um novo produto, cujo projecto será desenvolvido por uma empresa de consultoria, a qual será seleccionada no mercado.

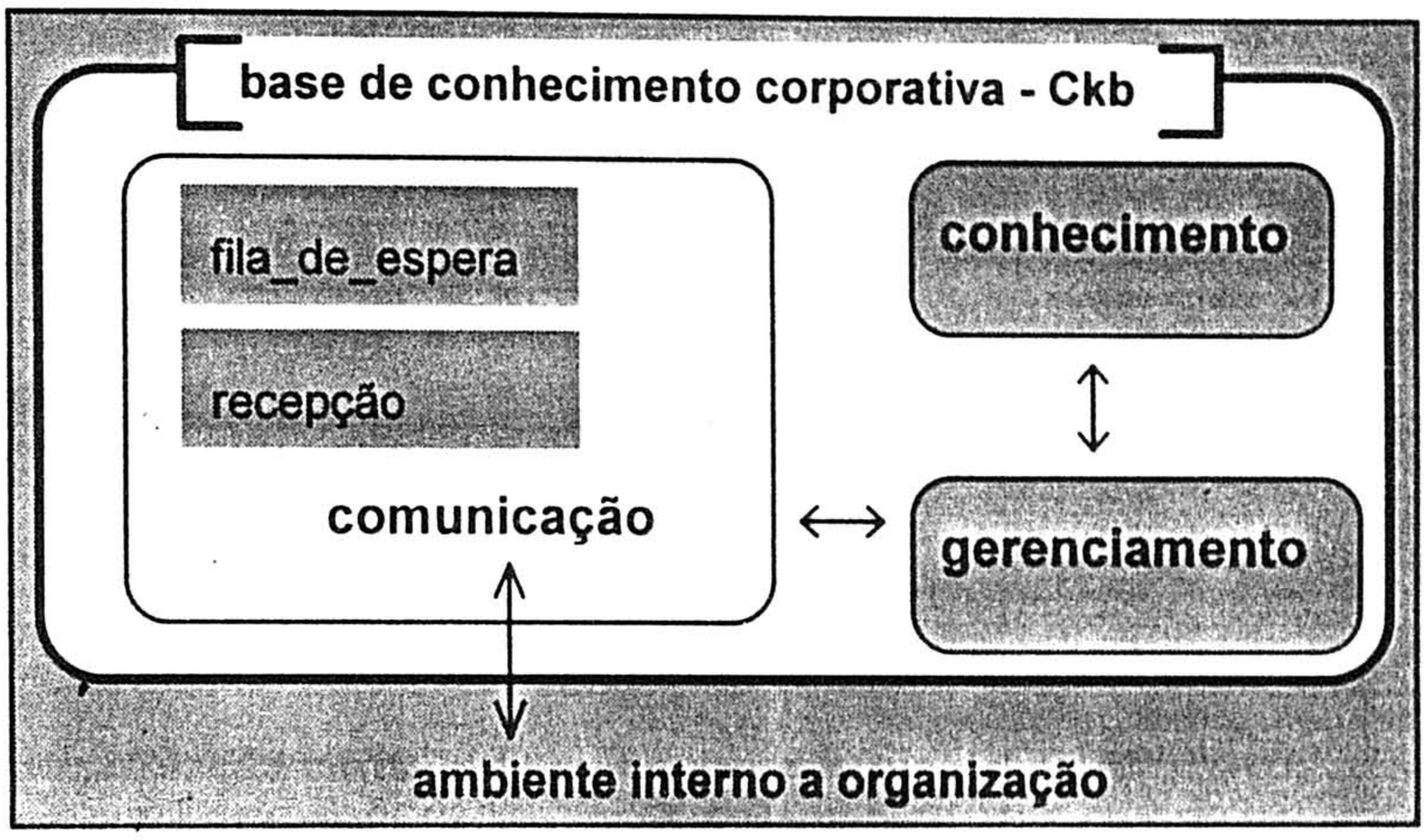
A base de conhecimento corporativa se apresenta com as seguintes capacidades [fig. 55]:

(i) comunicação, e

(ii) gerenciamento da base de conhecimento, para:

(ii.i) manter os conhecimentos, a pedido, gerado pelos agentes que compõem um agente envolvente, e

(ii.ii) responder a perguntas, sob solicitação, dos agentes que compõem um agente envolvente.



[fig.55] Capacidades da base de conhecimento corporativa

A base de conhecimento corporativa é lançada a exemplo dos demais agentes, cuja especificação dos módulos que implementam as capacidades são referidos na própria base:

```
/*******************************************************/
i_am( artor_org_A_corporate_knowledge_data_base).
/*******************************************************/
% Self Model - modelo do auto conhecimento
/*******************************************************/
is_part_of( managing_knowledge_capacity,
  artor_org_A_corporate_knowledge_data_base).
is_part_of( communication_capacity,
  artor_org_A_corporate_knowledge_data_base).
% The communication_module
is_a( artor_comm_m1, communication_capacity,
  artor_org_A_corporate_knowledge_data_base, 1).
% The managing_knowledge_module
is_a( artor_makn_m1, managing_knowledge_capacity,
  artor_org_A_corporate_knowledge_data_base, startded_by).
%
/*******************************************************/
```

Quando um agente interno a um agente envolvente, solicita um conhecimento à base de conhecimento corporativa, este agente interno pode reservar o conhecimento, se:

(i) tiver permissão para tal (será visto no item "A coordenação"), e

(ii) o conhecimento apresentar a característica de poder ser reservado.

Como exemplo, a actual posição dos recursos financeiros é solicitada por um agente, para verificar, como pré-condição, a possibilidade de disparar uma actividade ou tarefa contida nos planos. Para tal, o agente envia a mensagem de pedido do conhecimento com a opção ("*locked*"). Após o recebimento da mensagem resposta e a verificação, por parte do agente solicitante, caso seja suficiente a quantidade de recursos, é enviada uma mensagem de actualização do conhecimento e o respectivo desbloqueio ("*unlocked*") para a base de conhecimento corporativa, caso contrário apenas é enviada a mensagem solicitando o cancelamento da reserva.

```
/*************************************************************/
% Corporate_Knowledge - "artor_org_A"
/*************************************************************/
% Self Model - Sobre recursos
/*************************************************************/
%    resource( Type, Value, Status)
%    onde: Status = {locked, unlocked}
%
resource( recursos_finaceiros, 30000, unlocked).
...
%
```

A opção "*locked*" assegura, por parte de um agente, que durante a verificação das pré-condições e a alocação de alguns recursos, não percam a integridade do conhecimento, caso um outro agente necessite dos mesmos recursos.

Para que os agentes possam aceder, de forma a não usufruirem de previlégios frente a outros, quanto a obtenção dos conhecimentos que apresentem a característica de poderem ser reservados, o seguinte comportamento é apresentado pela base de conhecimentos, a cada mensagem recebida via predicado de leitura ("*read_socket*"):

(i) se a mensagem solicita reserva de conhecimento, então esta é colocada na fila de espera, caso contrário é atendida,
(ii) a cada entrada da fila de espera, é verificada a permissão do agente em reservar o recurso e o estado de reserva deste,
(iii) se ambos os critérios forem satisfeitos, a mensagem recebida é retirada da fila, o conhecimento solicitado tem seu estado de reserva alterado e a mensagem resposta é enviada ao agente solicitante,
(iv) caso o critério de reserva não esteja satisfeito, a mensagem é deixada na sua posição relativa de chegada na fila de espera, e uma mensagem informativa é enviada ao agente solicitante,
(v) caso apenas o critério de permissão não esteja satisfeito, uma mensagem é enviada contendo a informação de não permissão.

2.5.3.2.3 A coordenação

A versão que implementa a capacidade de coordenação nos agentes, é a referida na base de conhecimento corporativa, cujo módulo é o "*artor_coor_c1*". Para exercer a competência de coordenação no domínio do planeamento global, os agentes gestores implementam três metodologias de coordenação:

(i) a padronização,
(ii) a supervisão directa, e
(iii) o ajuste mútuo através da argumentação.

O uso da padronização ocorre, quando o agente gestor, baseado no descritor de modelos de permissão a seguir apresentado, obtém da base de conhecimento corporativo as ocorrências do planeamento global de sua competência. Os planos são vistos como padrões a serem seguidos, orientadores de comportamentos. Os predicados a seguir definem o descritor para o agente " *artor_org_A_director*" relativo a sua capacidade de coordenação. Neste exemplo a capacidade do agente permite recuperar e actualizar ("*update*") todos os níveis do planeamento global e em todas as suas ocorrências("*[up_to, op, all]*"). Os modelos de permissão enquadram a visão de um agente gestor, em função de suas responsabilidades que o mesmo comprometeu frente a uma organização (agente envolvente).

```
/*****************************************************/
% Corporate_Knowledge of the agent "artor_org_A"
/*****************************************************/
% descriptor_of_the_permissions
/*****************************************************/
% permission(Type, Modality,
%              [Permission_attributes], Capacity, Agent).
%
permission(plans, update, [up_to, op, all],
        coordination_capacity, artor_org_A_director).
/*****************************************************/
```

A supervisão directa ocorre quando o agente gestor, delega em um agente subordinado a actividade à ser realizada ou a tarefa à ser executada, e supervisiona-a até o encerramento da mesma, seja este encerramento com sucesso ou não.

O ajuste mútuo ocorre quando o agente ao qual está sendo delegada a tarefa, não tem possibilidade de cumpri-la, por exemplo no prazo desejado. Isto ocorre em função do mesmo estar já compromissado com outras, não podendo, no caso, as demais serem postergadas. Para esta situação:

(i) procura-se, internamente ao agente envolvente, outro agente com a mesma competência, se existir, para atribuir-lhe a actividade ou tarefa, ou

(ii) caso não exista outro agente com competência para sua execução, há necessidade de replaneamento da mesma, seja a nível de contratação de um novo recurso, seja no remanejamento do período de execução.

Baseados nestas metodologias, a coordenação para a realização do planeamento global deve satisfazer :

(i) a obtenção dos planos,

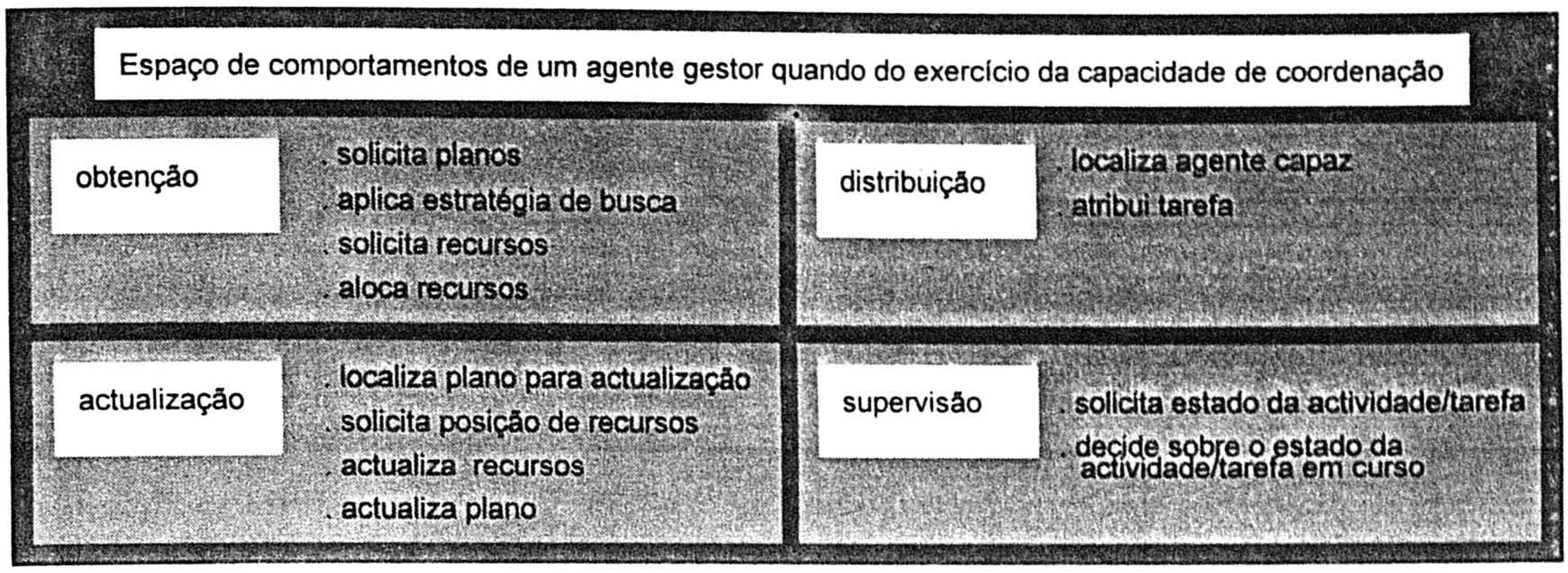
(ii) a distribuição das actividades e tarefas aos agentes competentes,

(iii) a supervisão das actividades e tarefas, e

(iii) a actualização das ocorrências do plano.

O espaço dos comportamentos [fig. 56], que suportam a capacidade de coordenação, são modelados pela tripla:

```
action( comportamento_activado,
        mensagem_recebida,
        próximo_comportamento_a_activar).
```



[fig. 56] Espaço de comportamentos de um agente gestor para o exercício da capacidade de coordenação

O "*comportamento_activado*", no domínio da coordenação para realização do planeamento global, situa o estado do mundo através das variáveis que manipulou. A "*mensagem_recebida*" apresenta-se como o motivador do "*próximo_comportamento_a_activar*", considerando o estado do mundo.

Exemplo:

```
/*******************************************************/
% The IK - Individual Knowledge of the agent
%        "artor_org_A_director"
/*******************************************************/
% Ca - knowledge about coordination_capacity
/*******************************************************/
action( updating_plans,
  result_of_update_plans( Capacity, Status),
  getting_plans).
...

action( looking_for_task,
  has_task_found( Capacity, Status),
  triggering_task).
...
/*******************************************************/
```

As mensagens trocadas intra-agente envolvente no exercício da capacidade de coordenação, as quais são uma das componentes para a orientação de seus comportamentos no domínio em questão, são:

(i) de um agente gestor para uma base de conhecimento corporativa:

```
send_plans( Capacity, Plan_number).
send_subject_responsability( Capacity, Class_service).
update_plans( Capacity, [Update_plans]).
send_resources( Capacity, [[Modality,Resource],…]).
update_resources( Capacity, [ Resources_to_be_updated]).
```

onde:
Capacity é a capacidade do agente que solicita o conhecimento, por exemplo: coordination_capacity,
Plan_number = {identificador da ocorrência, _},

Class_service é a classe do serviço (actividade/tarefa) à ser realizada em um dos sub-domínios de actuação do agente envolvente,
Update_plans = [[Type, identificador da ocorrência,
Status_plan], ...], e
Modality = {read_only, update}.

(ii) de uma base de conhecimento corporativa como resposta a solicitação de um agente gestor:

plans(Capacity, Plans, Status).
subject_responsability(Capacity, [Class_service, [Agent_able, PortNumber],…], Status).
result_of_update_plans(Capacity, Status).
resources(Capacity, [[Modality, Resources, Status], …]).
result_of_update_resources(Capacity, Status).

onde:
Status_for_plans {ok, no_plans, not_authorized},
Status_for_subject_of_responsability{ok, no_agent},
Status_for_update_plans = {successful, not_successful, not_authorized},
Status_for_resources ={locked, ok, not_authorized}
locked é a informação que o conhecimento está reservado, porém o pedido encontra-se na fila de espera,
Status_for_update_resources = {successful, not_successful, not_authorized}.

(iii) entre agentes gestores, e agentes gestores e executores:

(iii.i) para solicitação de execução de um serviço (actividade/tarefa):

execute_service(Capacity, Type, Identificação_da_ocorrência, Class_service, To_Agent).

(iii.ii) resposta a solicitação de execução de um serviço (actividade/tarefa):

execute_service(Capacity, Type, Identificação_da_ocorrência, Class_service, Status).

onde:

Status ={accepted, not_accepted}

(iii.iii) para solicitação sobre o estado de um serviço (actividade/tarefa):

status_service(Capacity, Type, Identificação_da_ocorrência, To_Agent).

(iii.iv) resposta a solicitação sobre o estado de um serviço (actividade/tarefa):

```
status_service(                    Capacity,                    Type,
Identificação_da_ocorrência,
 Status).
```

onde:
Status ={w, i, a}
w = a espera,
i = em andamento,
a = realizado,
n = não realizado.

(iv) de existência, a qual é enviada uma única vez pelos agentes intra ou inter agente envolvente, por ocasião do primeiro contacto, independente do tipo de relação social que será estabelecida. Esta mensagem tem por finalidade, garantir a existência do agente. Caso não haja resposta, este estado orientará o comportamento a ser adoptado por parte do agente emissor.

```
are_you_there.
```

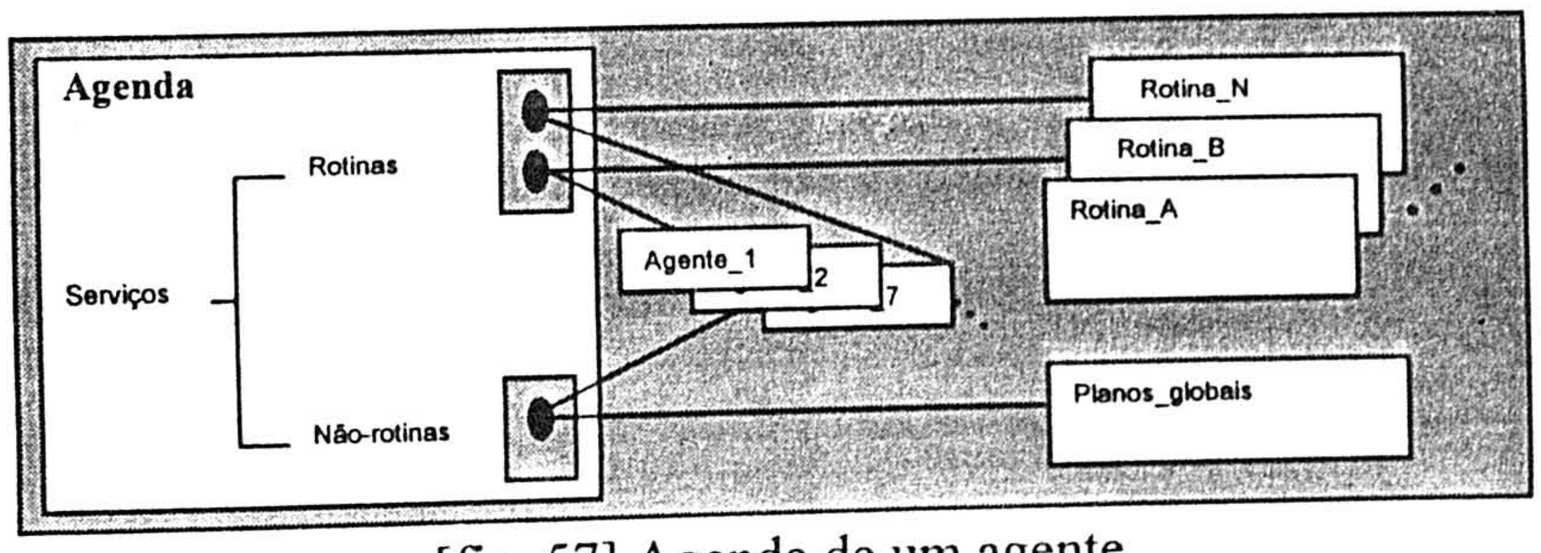
e cuja resposta, se houver, será:

```
are_you_there( yes_I_am).
```

Os agentes gestores, durante seus ciclos de existência, mais especificamente nos ciclos do exercício das capacidades de coordenação estão, baseados em suas agendas (contidas no conhecimento individual - "*IK*") [fig. 57] coordenando:

(i) os serviços rotineiros, os quais são partes de suas competências nos sub-domínios do agente envolvente, e

(ii) os serviços não rotineiros, os quais foram solicitados pelos agentes aos quais estão subordinados pela autoridade, ou não, dependendo do tipo de estrutura organizacional ("*his*" ou "*hiw*").



[fig. 57] Agenda de um agente

```
/*********************************************************/
% The IK - Individual Knowledge of the agent
%        "artor_org_A_director"
/*********************************************************/
% Sd - knowledge about sub-domain
% agenda
/*********************************************************/
service( Service_type, Service).
service_status( Service, Related_to, Agent, Status).
```

```
routine(Routine_identification, Attributes_of_routine).
/*************************************************************/
onde:
  Service_type = (routine, adhoc),
  Service é a identificação do serviço,
    se routine contém o identificador da rotina à ser
         executada,
    se adhoc:
         contém os atributos de uma ocorrência de um plano
         global, os quais são:
                tipo do nível da ocorrência = (ob, es, ta,
         op),
                identificador da ocorrência,
Related_to para o caso de routine, é o indicativo de qual
 período, por exemplo se período é semanal, contém a semana
 específica,
Agent = [Agent_name, Agent_PortNumber]
  é o agente designado para sua execução,
Status = (w, i, a, n)
       w = ocorrência em espera,
       i = ocorrência em realização,
       a = ocorrência realizada,
       n = ocorrência não realizada,
Attributes_of_routine  é uma lista que contém os atributos:
              período de execução = (diário, semanal, ...),
              descrição do serviço.
```

Cada ocorrência de um plano global deve ser atribuida aos agentes que são responsáveis pelos subdomínios da organização. Estas ocorrências consistem de serviços (actividade/tarefas) não rotineiras. A realização de um plano global, é iniciada pelo(s) agente(s) que representam a camada mais alta da dimensão horizontal. Os agentes gestores das demais camadas horizontais, não tomam a iniciativa, pois dependem de uma ordem de um nível superior.

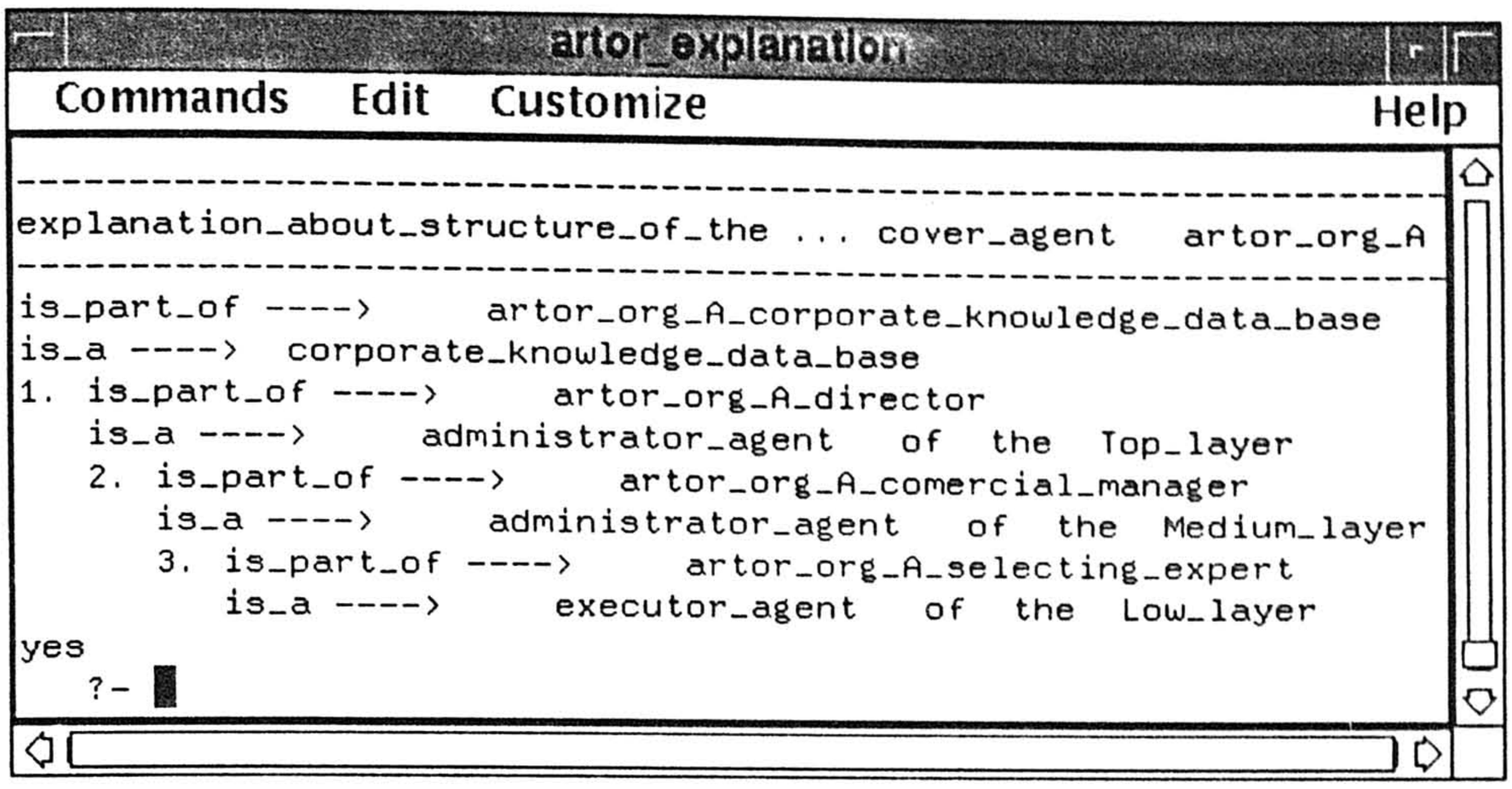
Os agentes que pertencem a camada de mais alto nível da dimensão horizontal são obtidos com base no predicado (contido na base de conhecimento corporativa):

$$is_part_of(\ Agent_1,\ Agent_2).$$

quando o agente ($Agent_1$) é sub-parte primeira do agente envolvente($Agent_2$).

Por exemplo, se interrogado via "*artor_human_being_machine_interface_status*", qual a estrutura do agente envolvente "*artor_org_A*", obteremos a seguinte resposta [fig. 58]:

BIBLIOTECA F.E. PORTO



[fig. 58] Explicação sobre a estrutura de um agente envolvente

O(s) agente(s) gestor(es) da camada mais alta da dimensão horizontal solicita(m) à base de conhecimento corporativo os planos globais, sem a identificação de uma ocorrência específica:

```
send_plans( Capacity, _).
```

Para os agentes gestores das demais camadas horizontais, esta solicitação de planos à base de conhecimento corporativa, passa a ser instanciado com a identificação da ocorrência que lhe foi ordenada.

```
send_plans( Capacity, Plan_number).
```

A base de conhecimento corporativa, através de sua capacidade de gerenciamento, utiliza o predicado:

```
permission( Type, Modality,
        [Permission_attributes], Capacity, Agent).
```

para, enviar ao agente solicitante, os planos correspondentes a sua alçada:

```
plans( Capacity, Plans, Status).
```

Havendo planos ("*Status* = *ok*"), e com base no predicado,

```
has_responsible_for( Agent, Subject_of_responsability).
```

o qual foi carregado para a base de conhecimento individual, a partir da base de conhecimento corporativa em tempo de lançamento do agente, o agente gestor aplica para cada ocorrência de sua competência, as seguintes estratégias de busca, para seleccionar as ocorrências à serem disparadas para os demais agentes do agente envolvente:

(i) a ocorrência cujo estado for a espera ("w ≐ wait"),

(ii) a ocorrência que contiver a mais alta prioridade (o menor numeral corresponde a mais alta prioridade),
(iii) caso todas as ocorrências apresentem a mesma prioridade, os agentes seleccionam aquelas cujas distâncias entre a data corrente e a data prevista para início (data de início é uma das pré-condições), for maior ou igual a zero. No sentido da distância maior para a menor,
(iv) a ocorrência que tiver o menor período de execução (utiliza a data limite para a realização).

Para cada ocorrência seleccionada, o(s) agente(s) gestor(es) verificam as pré-condições. As pré-condições são módulos carregados em tempo de execução, e estão descritos no "*descritor_dos_artefactos*" a exemplo dos demais componentes do sistema "*ARTOR*". As pré-condições, podem ser por exemplo, relativas a disponibilidade de recursos. Se for este o caso, o agente envia uma mensagem a base de conhecimento corporativa, solicitando este conhecimento:

```
send_resources( Capacity, [[Modality, Resources],...]).
```

A base de conhecimento envia os recursos através de:

```
resources( Capacity, [[Modality, Resources, Status], …]).
```

Caso o estado dos recursos seja "*ok*", o agente passará a verificar quanto as pré-condições impostas. Se todas forem satisfactórias, enviará a base de conhecimento corporativa, a solicitação dos agentes que são capacitados para executarem o serviço:

```
send_subject_responsability( Capacity, Class_service).
```

A base de conhecimento responde a solicitação:

```
subject_responsability(Capacity, [Class_service, [Agent_able,
  PortNumber],…], Status).
```

Caso haja apenas um agente capaz de executar o serviço, o agente coordenador verifica se o mesmo é seu subordinado hierárquico, para que possa ordenar-lhe o serviço. Caso contrário a argumentação seria utilizada para designa-lo. Para esta verificação, o predicado "*is_part_of*($Agent_m$, $Agent_k$)" é utilizado. Por exemplo para a organização "*artor_org_A*":

```
is_part_of( artor_A_comercial_manager, artor_A_director).
```

Para um agente virtual, o predicado é:

```
is_virtual_part_of(Agent_m, Agent_k).
```

Caso existam mais que um agente capaz, o coordenador verificará se existe algum conhecimento registado em sua base de conhecimento individual que permita avaliar qual dos agentes possuem qualidades sobre os demais, por exemplo; cumprir os prazos estabelecidos.

Caso não exista nenhum, o agente coordenador procura a próxima ocorrência do plano, pois a actual ou necessita ser replaneada ou um agente ser contratado.

Para ordenar o serviço, a seguinte mensagem é enviada ao agente escolhido:

```
execute_service( Capacity, Type, Identificação_da_ocorrência,
  Class_service).
```

O potencial agente capaz responderá com o aceite ou não, dependendo de sua capacidade de realização,

```
execute_service(                    Capacity,              Type,
Identificação_da_ocorrência,
  Class_service, Status).
```

a qual é estimada da seguinte forma:

```
  Tetcap_periodo - Tescap_periodo = Tpot

onde:
  Tetcap_periodo é o tempo total estimado para o exercício de
    sua capacidade no período para o qual o serviço está
  sendo
    ordenado,
  Tescap_periodo é o tempo total estimado já comprometido de
    exercício de sua capacidade no período,

  Se Tpot  ≥  Tesor
        então Estado = aceito ("accepted"),
        senão Estado = não aceito ("not_accepted").

    Tesor é o tempo estimado para o exercício de sua capacidade
      para o serviço que está sendo ordenado.
```

Se o serviço não for aceito ("*Status = not_accepted*"), o agente coordenador procederá a busca de um outro agente capaz, caso exista. Caso não haja mais agentes capazes, recai no caso da inexistência de agentes.

Se o serviço for aceito ("*Status = accepted*"), o agente coordenador procede a actualização:

(i) de sua agenda, relativo ao serviço e ao agente para o qual este serviço foi atribuído, e

(ii) dos recursos e dos planos na base de conhecimento corporativa:

```
update_plans( Capacity, [ Update_plans]).
update_resources( Capacity, [ Resources_to_be_updated]).
```

e a base de conhecimento corporativa deverá confirmar os pedidos de actualizações:

```
result_of_update_plans( Capacity, Status).
result_of_update_resources( Capacity, Status).
```

As actualizações de um plano, podem implicar na alteração dos estados de outros, i.e. se um agente altera o estado de um plano táctico ("*ta*") do estado "em andamento" ("*in-progress*"), para o estado "alcançado" ("*achieved*"). O agente responsável pelo plano estratégico ("*es*") que contém o táctico referido, deve altera-lo para "alcançado", caso todos os demais planos tácticos já tenham sido executados.

Este comportamento se apresenta, sem a troca de mensagens, sendo o agente o responsável por manter actualizado o estado do nível do plano que lhe foi atribuído.

Quando um serviço é encerrado, os recursos utilizados devem ser actualizados. Para que as actualizações possam ocorrer há que se fazer distinção entre duas classes de recursos existentes:

(i) recursos retornáveis. São os recursos utilizados na execução de um serviço, porém tornaram-se disponíveis para serem utilizados em outros, e

(ii) recursos não retornáveis. São os recursos que foram utilizados e consumidos na execução de um serviço.

Por exemplo a alocação de um equipamento, após o encerramento do serviço, retorna como equipamento disponível. Um recurso financeiro o qual não foi totalmente consumido, retorna apenas de forma parcial.

Na supervisão dos serviços distribuídos, os agentes gestores, a partir de suas agendas, solicitam aos agentes, para os quais foram atribuídos os serviços, qual o estado dos mesmos:

```
status_service( Capacity, Type, Identificação_da_ocorrência,
  To_Agent).
```

Os agentes que recebem o pedido, deverão informar o agente que solicitou a informação, qual o estado dos serviços a eles atribuídos:

```
status_service( Capacity, Type,  Identificação_da_ocorrência,
 Status).
```

Os agentes que coordenam os serviços em execução, a partir dos estados informados, compara-os com os atributos de data prevista de início, data prevista de término e tempo de execução, que estão disponíveis em suas agendas. O resultado destas comparações leva os agentes gestores a decidirem se:

(i) permanecem os serviços em seus estados,

(ii) cancelam os serviços, ou

(iii) replaneiam-os.

2.5.4 Os componentes do sistema "*ARTOR*" - O quiosque "*News-Stand*"

O quiosque "*News-Stand*" representa um local de afixação pública de informações para os componentes de uma sociedade de organizações, possibilitando que os agentes interessados na obtenção de notícias possam, neste local, obtê-las. Sua implementação computacional ocorre através da utilização de um quadro-negro, apresentando as seguintes capacidades:

(i) receber as mensagens provenientes dos agentes envolventes ou dos indivíduos. Estas mensagens são notícias no domínio das relações sociais,

(ii) disponibilizar as notícias para os agentes envolventes ou indivíduos, interessados no domínio das mesmas, e

(iii) coordenar a veiculação dessas notícias, quanto a:

(iii.i) quantidade em que a mesma que deve ser disponibilizada. A exemplo da venda de jornais, a qual encerra quando esgota os exemplares disponíveis, as notícias disponibilizadas possuem esta mesma característica,

(iii.ii) período de validade da notícia (início e término),

(iii.iii) agentes da sociedade que já a elas acederam.

O quiosque "*News-Stand*", implementado através do módulo "*artor_stand_news1*" especificado no descritor_do_artefacto_"*artor*", é lançado a exemplo do lançamento dos agentes envolventes.

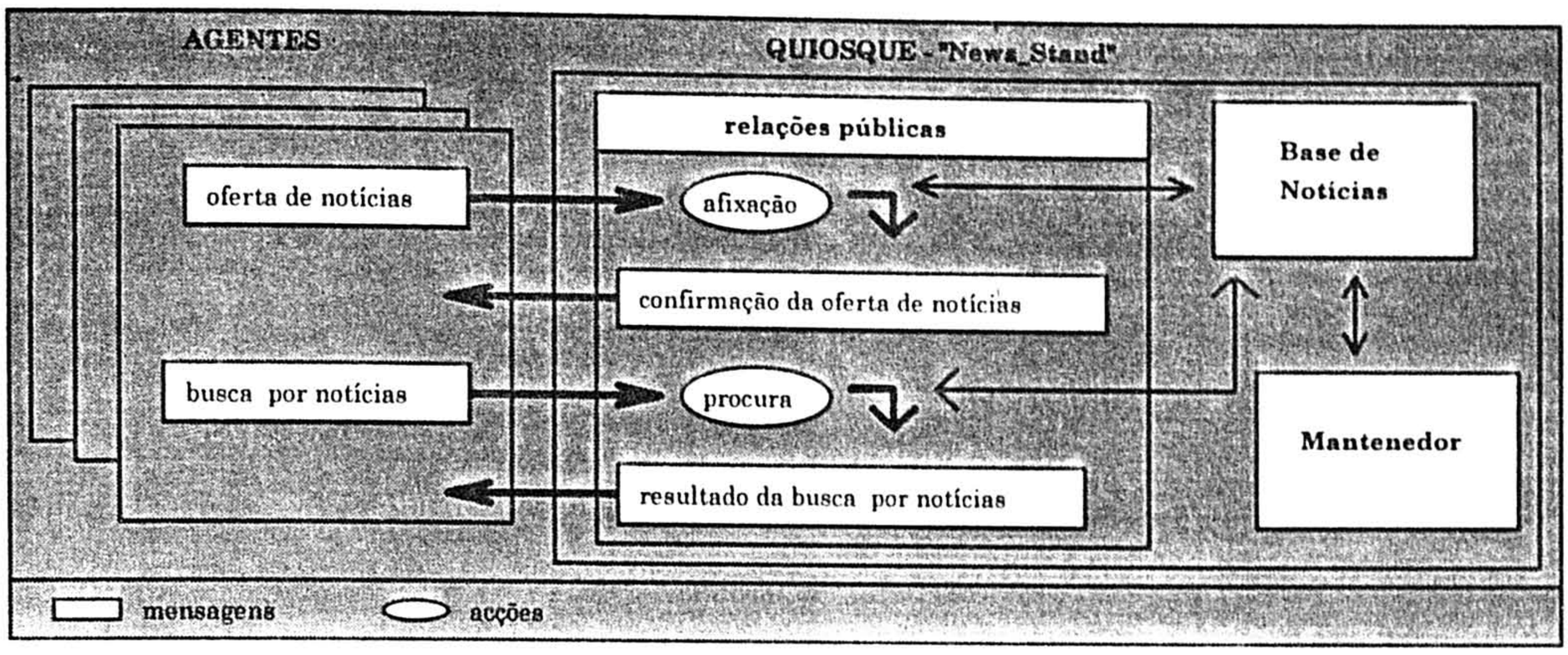
O procedimento principal do quiosque "*News-Stand*" é composto pelos seguintes predicados "*YAP*":

```
quiosque:-
        % identificação do quiosque, conforme descritor_do_artefacto_"artor"
  i_am( Quiosque),
        % obtenção da identificação da estação de trabalho
  host_name( Host_name),
        % procedimento de criação do suporte computacional para as comunicações
  creating_communication_support( Quiosque, Host_name),
        % ciclo que constitui a existência do quiosque
  quiosque_life( Quiosque, Host_name).
```

```
/*****************************************************************/
%                 *********  quiosque_life *********
/*****************************************************************/
quiosque_life( Quiosque, Host_name):-
  repeat,
        % leitura das mensagens provenientes dos componentes da
        % da sociedade e tratamento das mesmas
  public_relations,
        % manutenção do quiosque
  news_maintenance,
  fail.
```

/**/

Três são os componentes do quiosque "*News-Stand*" [fig. 59]:



[fig. 59] Arquitectura do quiosque "*News-Stand*"

(i) um relações públicas, cujas actividades são:

(i.i) receber as mensagens provenientes dos agentes da comunidade,

(i.ii) activar os comportamentos relativos as mensagens, executando as acções de afixação ou de procura de notícias, e

(i.iii) responder as mensagens recebidas,

(ii) um mantenedor da base de notícias o qual providencia a retirada de notícias quando as mesmas perderam a válidade para veiculação, seja por período ou por quantidade, e

(iii) uma base de notícias, a qual mantém o registo das mesmas, em função do operador emitido pelos comportamentos.

Quando as mensagens são recebidas pelo quiosque, mais especificamento pelo componente "relações públicas", provenientes dos agentes da sociedade, elas são atendidas na ordem de chegada ("*fifo*"), não dando preferência, em particular, a nenhum dos agentes.

As mensagens são:

(i) de pedido de afixação de notícias,

```
news_offer( news_id( News_id),
            news_subject( News_subject),
            news_content( News_content),
            news_deadline( News_deadline),
            news_boundary( Boundary_type, Boundary_value)
            ).
```

onde:

"*News_id*" é a identificação da notícia à ser afixada,

"*News_subject*" é o título da notícia, ou

"*cancel*", no caso de exclusão da notícia a pedido. Para esta situação os demais argumentos não são necessários

"*News_content*" é o conteúdo da notícia,

"*News_deadline*" é a data limite para resposta, por parte dos agentes nela interessados e caso a notícia requeira uma data de validade. Ex: promoção válida até ...

"*news_boundary*" é o limite para veiculação da notícia i.e. se a notícia estará disponível até uma certa data e/ou em uma certa quantidade.

"*Boundary_type*" = {data, quantidade}

"*Boundary_value*" é o valor correspondente ao tipo

e cuja resposta é:

```
comfirming_news_offer( news_id( News_id),
        news_subject( News_subject),
        news_acknowledgment( News_acknowledgment)
        ).
```

onde:

News_acknowledgment = {accepted, not_accepted}

(ii) de solicitação de notícias,

```
asking_for_news( news_subject( News_subject)
                ).
```

e cuja resposta é:

```
(i)answering_ask_for_news( news_id( News_id),
        news_content( News_content),
        news_deadline( News_deadline),
        news_from_agent( Agent_name, Hostname, Portnumber)
        ).
```

onde:

"*news_from_agent*" é a identificação do agente que disponibilizou a notícia

```
(ii)answering_ask_for_news(
        that_is_all( News_subject)
        ).
```

onde:

"*that_is_all*" informação de final de envio de notícias solicitadas

```
(iii)answering_ask_for_news(
        there_is_no_news_offer_for_the( News_subject)
        ).
```

A base de notícias contém dois grupos de conhecimento:

(i) as notícias propriamente dita, e

(ii) os agentes que as acederam.

Estes conhecimentos são, respectivamente registados nos seguintes formatos:

(i) as notícias propriamente ditas:

```
news_offer( news_id( News_id),
        news_subject( News_subject),
        news_content( News_content),
```

```
news_deadline( News_deadline),
news_boundary( Boundary_type, Boundary_value),
news_from_agent( Agent_name, Hostname, Portnumber),
registration (Resgistration_time)
).
```

onde:

"*Registration_time*" é a data na qual a notícia foi afixada

(ii) os agentes que as acederam:

```
news_reached( News_offer_rec_number,
interested_agent( Agent_name, Hostname, Portnumber),
registration( Registration_time)
).
```

onde:

"*News_offer_rec_number*" é o número único atribuido pela base de notícias quando do registo de notícias na mesma.

Relativo aos comportamentos apresentados pelo componente "relações pública", este aquando das mensagens:

(i) de oferta de notícias para afixação,

executa:

(i.i) a verificação quanto a sua inexistência na base de notícias,

(i.ii) a verificação de validade, nos domínios, dos argumentos: "*News_deadline*" e "*Boundary_type, Boundary_value*",

(i.iii) o registo na base de notícias, e

(i.iv) o envio da mensagem para o agente solicitante da afixação, contendo o aceite ou o não, caso as verificações apresentem-se inconsistentes,

(ii) de oferta de notícias para cancelamento,

executa:

(ii.i) a verificação quanto a existência da notícia,

(ii.ii) a exclusão da mesma da base de notícias, e

(ii.iii) envio da mensagem para o agente solicitante do cancelamento, contendo o aceite ou o não aceite, caso a verificação apresente-se insatisfactória,

(iii) de solicitação de notícias,

executa:

(iii.i) uma pesquisa sobre a base de notícias procurando aquelas que atendam ao assunto desejado, através do critério "*News_subject*",

(iii.ii) a verificação sobre as notícias seleccionadas, quanto aos critérios de veiculação (prazo ou quantidade) e sobre os registos dos agentes que já as tenham acedidas. Para os agentes que não as tenham acedidas, é procedido o registo de identificação do mesmo, vinculando-o à notícia, e

(iii.iii) o envio da mensagem para o agente solicitante, ou com as notícias resultantes da pesquisa ou com a informação de inexistência.

O componente "mantenedor da base de notícias" tem por função retirar as notícias cujo período de veiculação esteja satisfeito. Além de remover as notícias já não mais válidas, os agentes que as acederam, também são removidos.

2.6 Conclusões

O sistema "*ARTOR*", permite modelar uma sociedade de organizações, baseado em metodologias de Inteligência Artificial Distribuída, mais especificamente as de Sistemas Multiagente. O objectivo último do sistema "*ARTOR*" é guiar a implementação computacional do modelo das organizações, permitindo uma automatização de relações inter e intra organizações.

Uma sociedade de organizações, para o sistema "*ARTOR*" possui os seguintes componentes:

(i) as organizações,

(ii) um quiosque, local público de afixação de notícias, e

(iii) uma interface homem/artefacto.

As organizações são modeladas por agentes denominados de agentes envolventes.

Em um agente envolvente as competências para exercer as actividades requeridas pelos sub-domínios que constituem uma organização, são distribuídas em dois tipos de agentes:

(i) os dotados de competência para gerir a organização, chamados de agentes gestores, e

(ii) os dotados de competência para executar as especialidades requeridas pela organização, chamados de agentes executores.

Esta distribuição de competências permite explorar a execução das capacidades dos agentes de forma modular, estruturada, paralela e com diferentes graus de autonomia. Concentra a execução da especialidade nos agentes executores, modelando-os sem capacidades de planeamento e de coordenação de suas competências perante a organização. Por sua vez, os agentes gestores, dedicam-se à satisfação dos objectivos organizacionais, gerindo de forma integrada as competências das especialidades a eles vinculadas, num certo período.

O outro componente de um agente envolvente, a base de conhecimento corporativa, representa o conhecimento que não pertence, em particular, a nenhum dos agentes que compõem a organização. O conhecimento corporativo é adquirido, aprendido e mantido pelos agentes internos ao agente envolvente, em função das suas intra e inter relações sociais. O conhecimento corporativo é orientador do comportamento dos agentes gestores e executores no desempenho de suas competências, tendo como finalidade que as actividades a serem executadas estejam de acordo com os objectivos, intenções e perfil da organização.

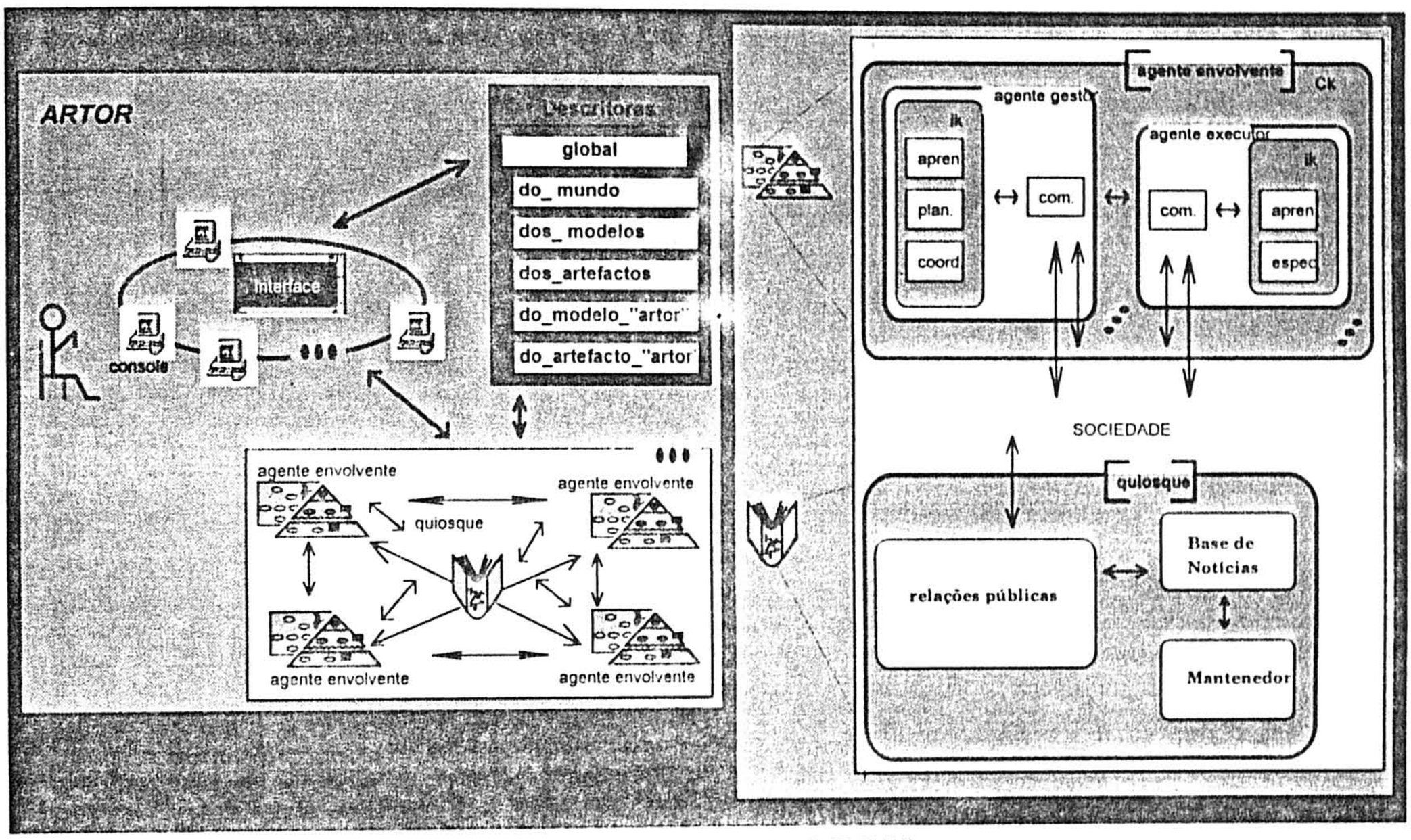
O quiosque, local público de afixação de notícias, tem por função disponibilizar conhecimentos sobre factos ou eventos sociais para os agentes envolventes que estão interessados em oportunidades de negócios. Sua implementação é baseada em metodologias de "quadro-negro", possibilitando a

afixação, de notícias, a gestão de suas veiculações e a disponibilização das mesmas quando solicitadas.

A interface homem/artefacto permite ao utilizador visualizar as trocas de mensagens entre os componentes intra e inter agente envolventes. Permite inquirir os agentes envolventes quer quanto as suas actividades quer sobre o seu conhecimento.

Para suportar a modelação de uma sociedade de organizações, baseada em recursos computacionais, que permita de uma maneira flexível e incremental implementar os componentes e as relações desta sociedade, o sistema "*ARTOR*" [fig. 60] utiliza:

(i) descritores parametrizados pelo utilizador, e
(ii) mecanismos genéricos de geração de componentes da sociedade.



[fig. 60] O sistema "*ARTOR*"

Os descritores, descrevem e estruturam o conhecimento sobre os modelos e artefactos relacionados com uma sociedade de organizações. São utilizados para a geração e para o lançamento dos componentes, bem como para suportarem as explicações no domínio.

Os mecanismos genéricos de geração de componentes, possibilitam que, de forma dinâmica, as capacidades e as metodologias que implementam as competências dos componentes sejam activadas.

Diferentes aproximações de modelação, no domínio das organizações [64, 66, 67, 68], recorrem a sistemas baseados em recursos computacionais, utilizando metodologias de Inteligência Artificial Distribuída, e/ou Sistemas Multiagente. O modelo apresentado por esta dissertação, buscou a generalização na construção de artefactos que suportem as diferentes estruturas e actividades organizacionais. Baseia-se na divisão das competências para a gestão e para execução de uma especialidade nos vários sub-domínios de actuação de uma organização, apresentando os seguintes benefícios:

(i) eficiência pelo partilhamento do tempo, permitido pelo paralelismo,

(ii) expansibilidade, possível pela agregação de novos agentes, seja na comunidade, seja intra agente envolvente,
(iii) modularidade, através do decréscimo de complexidade dos componentes intra agente envolvente,
(iv) compreensão de uma organização, possível pelo particionamento da competência global de uma organização e pela modularidade, e
(v) reutilização, pois os componentes podem ser utilizados na construção de outros agentes envolventes.

Quanto a distribuição, os componentes (agentes) do sistema "*ARTOR*" podem ser classificados:

(i) do ponto de vista intra-agente envolvente, os componentes são distribuídos e de controlo variando, entre a total descentralização e a total centralização, dependendo da estrutura organizacional adoptada, e
(ii) do ponto de vista da sociedade, os componentes são também vistos como distribuídos e de controlo descentralizado.

Capítulo 3

A RELAÇÃO SOCIAL, INTERORGANIZAÇÕES, DE SELECÇÃO E DE CONTRATAÇÃO DE COLABORADORES PARA O SISTEMA "*ARTOR*"

3.1 Introdução

As relações interorganizações para o provimento de bens/serviços, em uma sociedade de organizações, podem ser vistas como relações sociais de cooperação baseadas em compromissos, as quais contribuem no alcance dos objectivos organizacionais das partes envolvidas.

Numa sociedade de organizações, quando os componentes desempenham papéis de consumidores, significa que eles estão interessados em serem providos de bens/serviços. Quando os componentes desempenham papéis de distribuidores, significa que eles estão interessados em serem os provedores dos bens/serviços.

Para que estas relações se concretizem, uma das possibilidades, e a por esta tese abordada, é quando a iniciativa para o provimento de um bem/serviço é tomada pela organização que está desempenhando o papel de consumidora, sendo as demais possíveis fornecedoras.

Passaremos a chamar de **organizadores** as organizações que estão, neste tipo de relação, desempenhando o papel de consumidoras, e de **respondentes** as demais, que estão a desempenhar o papel de potenciais fornecedoras, caso expressem ao organizador os seus interesses em serem as provedores. Os termos, organizadores ("*organizers*") e respondentes ("*respondents*"), são referidos em [63], e são empregues no domínio da negociação onde respectivamente, para o caso, são os agentes que iniciam um processo de negociação e os que recebem o pedido para negociar.

O objectivo de um organizador, em uma relação de provimento, é adquirir ou contratar um bem/serviço. Como várias organizações na sociedade podem estar interessadas em proverem o bem/serviço desejado, o organizador deverá buscar a organização mais adequada. Para esta relação, ser uma organização mais adequada significa, a que atender os critérios de avaliação para escolha de fornecedores, definidos pelo organizador, em valores que satisfazem ou melhores que estes.

O objectivo dos respondentes, é o de se tornarem os provedores do bem/serviço. Os respondentes estão, através de seus comportamentos, interessados em

atender aos critérios estabelecidos pelo organizador, porém consideram os seus próprios interesses nas interacções da relação.

Os comportamentos dos respondentes, bem como o do organizador, são orientados pelo:

(i) conhecimento contido na base de conhecimento corporativo (*Ck*) de suas respectivas organizações, e

(ii) conhecimento contido no modelo de auto conhecimento(*Sm*) e no modelo dos agentes conhecidos (*Am*) de cada um dos agentes componentes de suas respectivas organizações.

Neste tipo de relação, um **organizador** apresenta os seguintes comportamentos:

(i) levar ao conhecimento da sociedade, o interesse em ser provido de um bem/serviço,

(ii) fazer propostas aos respondentes, contendo os critérios para avaliação e seus respectivos valores,

(iii) avaliar os respondentes, quanto as contrapropostas por estes formuladas,

(iv) detectar as situações de conflito i.e. caso mais de um respondente apresente contrapropostas que atendam ou não, os valores que satisfazem os critérios estabelecidos,

(iv) resolver os conflitos, e

(v) contratar um respondente, caso seja possível.

A este conjunto de comportamentos, por parte de um organizador, passaremos a denominar de **processo de selecção e de contratação de colaboradores para o provimento de um bem/serviço**. São colaboradores na perspectiva que colaboram, ou directamente através de suas competências/serviços ou através do bem por eles distribuídos.

Um **respondente** apresenta os seguintes comportamentos nesta relação:

(i) estar atento as organizações que buscam cooperação. A existência do interesse na procura de oportunidades de negócios é definida como actividade pertencente a um subdomínio da organização,

(ii) avaliar as propostas,

(iii) fazer contrapropostas.

As variáveis manipuladas pelos comportamentos das organizações, sejam elas organizadores ou respondentes, em uma relação de provimento de bens/serviços, são estados do mundo que compõem um conjunto de exemplos (atributos, valores), gerados pela experiência, sobre os quais as organizações podem aprender, no domínio da relação.

Este capítulo inicia apresentando a negociação como metodologia para resolução de conflitos e descreve alguns trabalhos no domínio da negociação automática. Após, analisa e propõe uma metodologia para implementar a relação social, entre organizadores e respondentes, de selecção e contratação de colaboradores, a qual emprega a aprendizagem simbólica automática para obter conceitos sobre o comportamento dos agentes envolvidos, no domínio da negociação. Esta relação é uma instância do sistema "*ARTOR*", onde as organizações comportam-se, através de seus agentes executores, como organizadores ou respondentes, dependendo do papel que estão a desempenhar, em função de seus planos e actividades.

3.2 Modelos para Suporte ao Processo de Negociação

A negociação é, fundamentalmente, um processo de tomada de decisão, o qual é aplicado quando se apresenta uma situação de conflito, durante as actividades para a realização de um objectivo comum, entre dois ou mais componentes de uma sociedade.

A negociação tem sido abordada em várias disciplinas [96, 97], tais como a teoria dos jogos, a economia , as relações internacionais e laborais e a psicologia. Ela tem recebido a atenção de investigadores nos domínios da Inteligência Artificial Distribuída [97, 98, 99, 100], bem como no desenvolvimento de modelos teóricos no qual a aprendizagem desempenha um importante papel.

Para os modelos de negociação, baseados em **teoria dos jogos** [95], parte-se do pressuposto que os elementos participantes comportam-se de maneira racional. O racional se apresenta, em função que a avaliação das alternativas baseiam-se nas preferências dos participantes, as quais são ordenadas de forma coerente e transitiva. Se o valor V_1 é preferível a V_2 e V_2 e preferível a V_3, então V_1 é preferível a V_3. A teoria dos jogos é fundamentalmente aplicada aos jogos chamados de jogos de soma zero, onde os ganhos de um participante, são possíveis através das perdas do outro. A inserção desta condição restritiva, reduz o processo de negociação, elimina as interacções estratégicas e simplifica o comportamento dos participantes. Um modelo experimental da teoria dos jogos é a chamada matriz de retribuição [fig. 61], na qual cada um dos participantes deve escolher, entre duas alternativas. Os resultados são apurados pela intersecção da escolha de um participante com a do outro.

		participante P_1	
		opção_a	**opção_b**
participante P_2	**opção_a**	V_1 de P_2, V_11 de P_1	V_2 de P_2, V_22 de P_1
	opção_b	V_3 de P_2, V_33 de P_1	V_4 de P_2, V_44 de P_1
V_x = valor da alternativa x de P_y			

[fig. 61] Matriz de retribuição

Se tomarmos o seguinte exemplo:

V_1=0, V_11=0	V_2=-50, V_22=50,
V_3=100, V_33=-100	V_4=25 e V_44=-25

Se *P_1* e *P_2* escolherem a *opção_a*, significa que não haverá negócio e consequentemente não haverá perdas nem ganhos de ambas as partes. Se *P_1* e *P_2* escolherem a *opção_b* haverá a menor perda e o menor ganho por parte dos participantes.

Frente ao exposto, observamos que:

(i) a soma dos valores contidos em cada célula é zero,

(ii) os participantes adoptam uma solução racional em função de um critério, o qual poderá ser o de minimizar as perdas máximas,

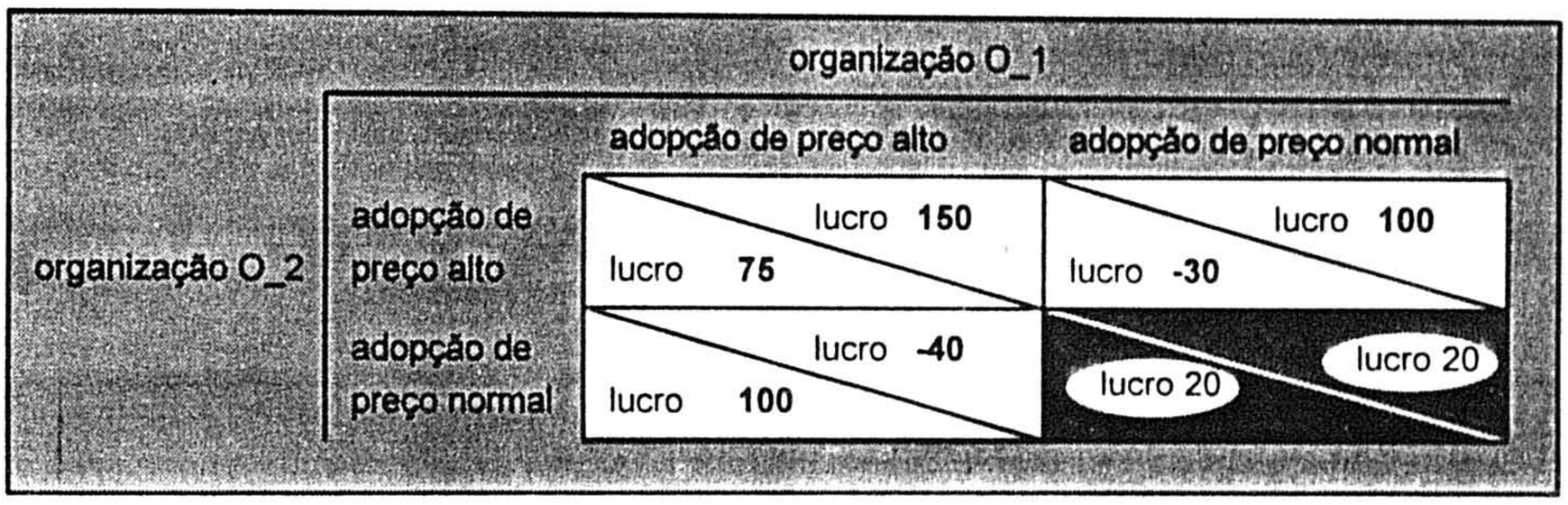
(iii) não há processo de negociação envolvido. As decisões são baseadas na presença de uma informação completa sobre a utilidade de cada uma das partes, e

(iv) o modelo exclui que a decisão seja tomada através de um acordo, via negociação.

Na teoria dos jogos, uma das questões importantes é pensar nos objectivos e acções do seu opositor e tomar a própria decisão com base neste raciocínio. Porém deve-se ter sempre em mente que, o opositor também irá comportar-se da mesma forma.

Ao considerar as estratégias possíveis, oriundas do raciocínio sobre os objectivos e acções do opositor, o caso mais simples é o de uma estratégia dominante. Esta é uma situação na qual um dos participantes tem a melhor estratégia, independentemente da estratégia adoptada pelos demais. No caso de dois participantes, quando os mesmos possuem uma estratégia dominante, diz-se que o resultado é um equilíbrio dominante.

Porém, a maioria das situações com interesse, não possui tal equilíbrio dominante, o que estão se designa por jogo da rivalidade. Como exemplo, no caso de um duopólio, cada uma das organizações deve considerar se adopta seu preço normal ou sobe-o até ao preço de monopólio, tentando auferir lucros advindos desta opção. Como exemplo, a matriz de um jogo de rivalidade [fig. 62], apresenta as seguintes considerações:



[fig. 62] Jogo da rivalidade

(i) se ambas as organizações (*O_1* e *O_2*) optarem por seguirem uma estratégia de preços elevados, ambas obterão o maior lucro conjunto (225). Esta situação caracteriza-se pela adopção de um comportamento monopolista,

(ii) se ambas optarem por manterem os preços normais, ambas auferirão lucros iguais (20, 20), caracterizando um comportamento concorrencial, e

(iii) se uma das outras duas outras opções for a escolhida, esta conduzirá à que a organização que mativer o preço normal, conquiste uma parcela maior do mercado e a outra obtenha prejuizo.

Neste exemplo e na perspectiva da organização *O_2*, está possui uma estratégia dominante, i.e. se optar pela adopção de preço normal, obterá lucro, independente da estratégia adoptada pela organização *O_1*. Já a organização *O_1* desejará optar por preço normal ou por preço alto, dependendo do que a organização *O_2* decidir.

Frente a este questão, a organização *O_1* encontra-se no dilema de colocar o preço alto e aguardar que a organização *O_2* a siga, ou decidir por uma posição segura, adoptando o preço normal. Como a organização *O_2*, possui uma estratégia

dominante, que é seguir a estratégia do preço normal, a organização *O_2* também a seguirá, pois nesta sua decisão encontra a melhor opção, baseada no princípio que os participantes agirão em seus próprios interesses e da melhor forma.

Este tipo de solução é chamado de equilíbrio de "*Nash*", também denominado de equilíbrio não cooperativo, o qual é caracterizado por nenhum dos participantes poderem aumentar o seus lucros frente a estratégia dos demais. No exemplo [fig. 62] o equilíbrio de "*Nash*" encontra-se no valores contidos na interseрção das adopções dos preços normais de ambas as organizações.

Então, dada duas organizações *O_1* e *O_2* cada uma adoptando uma estratégia *E_1* e *E_2* respectivamente. A dupla (*E_1* de *O_1*, *E_2* de *O_2*) é um equilíbrio de "*Nash*" se nenhuma organização puder encontrar uma estratégia melhor, admitindo que a outra mantenha sua estratégia original. Ou seja, enquanto *O_1* mantiver a estratégia *E_1*, *O_2* não poderá fazer melhor do que permanecer com a estratégia *E_2* e vice-versa.

Ainda no ambito da teoria dos jogos, aspectos com menor ênfase nos critérios racionais, são apresentados para o caso dos jogos de soma não nula, nos quais há razões não apenas para competir, mas também para existir uma certa colaboração entre os participantes. Um exemplo característico desta situação é dado pelo Dilema do Prisioneiro [fig. 63].

		suspeito S_1	
		não confessa	confessa
suspeito S_2	não confessa	2 anos, 2 anos	7 anos para S_2, 5 meses para S_1
	confessa	5 meses para S_2, 7 anos para S_1	5 anos, 5 anos

[fig. 63] Matriz para o Dilema dos Prisioneiros

Para o Dilema dos Prisioneiros, um comportamento racional, por parte de cada um dos suspeitos seria a confissão, porém se ambos a adoptarem, chegariam a um resultado inferior ao que obteriam se ambos não confessassem. O modelo apresentando vai um pouco além das teorias dos jogos, pois contribui para o estudo de processos de cooperação e de competição, presentes nos processos de negociação. Quando ambos actuam de forma a pensarem em seus próprios interesses, ou seja de forma não cooperativa, tem-se um resultado (confessa, confessa), o qual é um equilíbrio "*Nash*". Porém se optarem por uma estratégia cooperativa (não confessa, não confessa), podem evitar um período maior de reclusão.

Devido a algumas limitações apresentadas pela teoria dos jogos, para o processo de negociação, fundamentalmente por concentrarem-se nos resultados e dar pouca atenção aos processos que à eles conduzem, levaram a formulação de modelos dinâmicos. Uma negociação caracteriza-se pela dinâmica das propostas e contrapropostas, cuja atenção está centrada nos processos negociais, i.e. nas interacções entre os participantes. Se estes processos são estudados em uma perspectiva de decisões racionais, dão lugar aos chamados **modelos económicos** quando desenvolvidos no domínio desta disciplina, ou aos **modelos comportamentais**, estudados no âmbito da psicologia social. As diferenças existentes entre os modelos económicos e a teoria dos jogos reside nos seguintes aspectos:

(i) concentram-se em situações de soma não zero,
(ii) são dinâmicos, em relação a estática dos modelos baseados em teoria dos jogos, e
(iii) dão ênfase a formação de expectativas em relação ao comportamento dos participantes, o que por sua vez, na teoria dos jogos é suficiente conhecer os resultados alternativos para construir os critérios para acções racionais.

Mesmo frente a estas diferenças, os modelos económicos, devido a sua ênfase nos processos racionais, i.e. dedutivos, são levados a apresentarem também limitações, pois se abstraem de factores contextuais tais como os relativos aos recursos dos participantes ou as normas sociais.

Um dos modelos económicos experimentais é o chamado monopólio bilateral. O monopólio bilateral é a situação na qual um único consumidor encontra-se a frente de um único fornecedor. Neste modelo, os participantes devem chegar a uma acordo, por exemplo entre preço e quantidade para a troca de mercadorias, através de propostas e contrapropostas. Os participantes podem escolher entre diversas alternativas. Algumas dessas soluções correspondem ao chamado "*óptimo de Pareto*", o que significa a maximização dos lucros de ambos os participantes. Nem todos os pontos de uma curva correspondente ao "*óptimo de Pareto*", possuem um valor idêntico (ganhos iguais) para ambos os participantes. O processo de negociação envolvido neste tipo de modelo, consiste de duas etapas. A primeira está relacionada com a identificação dos valores que oferecem melhores resultados (curva que corresponde ao "*óptimo de Pareto*") para ambos os participantes. A segunda consiste em optimizar individualmente esses valores (ganhos).

A contribuição dos modelos económicos é fundamental na teoria das negociações, a medida que elabora uma estrutura sob a qual os participantes podem estabelecer as interacções. Porém as limitações do mesmo situam-se na não consideração dos factores que causam efeitos nos comportamentos, os quais os participantes podem recorrer, em uma negociação, para influenciarem as decisões dos outros.

Os paradigmas que acrescem à teoria dos jogos e aos modelos económicos, esses factores, são os baseados em:

(i) informações incompletas, e
(ii) estratégicas e tácticas,

para influenciarem os participantes nas suas tomadas de decisões.

Este paradigma será tratado, em uma perspectiva computacional, ao longo deste capítulo, pois a metodologia proposta para a implementação de uma relação de selecção e contratação de colaboradores, nele baseia-se.

3.3 Trabalhos Relativos a Automatização do Processo de Negociação

O recente crescimento pelo interesse em interacções entre agentes autónomos e suas potenciais aplicações no domínio do comércio electrónico [101], tem aumentado a importância pelos processos automatizados de negociação.

3.3.1 Modelo "*Bazaar*"

Motivados pela investigação de modelos computacionais de negociação suportados por agentes com capacidade de aprendizagem e que apresentam múltiplos critérios no processo de negociação, Zeng e Sycara [102] introduzem o modelo chamado "*Bazaar*". Para compreensão do funcionamento deste modelo, vejamos o seguinte exemplo:

(i) dado dois agentes (Ag_1 e Ag_2) os quais respectivamente desempenham os papéis de fornecedor e de consumidor de um certo bem,

(ii) Ag_2 é conhecido do agente Ag_1, e

(iii) assumindo que o canal de comunicação entre ambos está estabelecido,

Ag_1 necessita detalhar a sua oferta com respeito aos critérios de negociação, tais como o produto, o preço, a data de entrega, etc. A determinação das variáveis para estes critérios depende dos seguintes factores:

(i) estrutura de custos e de lucro,

(ii) conhecimento sobre a corrente situação económica do mercado,

(iii) a potencial procura deste produto, e

(iv) a expectativa que Ag_1 tem em relação Ag_2 quanto a tornar-se um potencial e lucrativo cliente em transações futuras.

Frente a estes factores, Ag_1 calcula o retorno esperado associado com as possíveis propostas que ele pode fazer e selecciona aquela que maximiza o retorno. Por sua vez Ag_2 recebe a proposta transmitida por Ag_1. Para decidir se aceita-a ou para fazer uma contraproposta, Ag_2 utiliza fundamentalmente um procedimento similar ao utilizado pelo Ag_1. Caso o Ag_2 decida aceitar a proposta, apenas necessita enviar uma mensagem ao Ag_1 com a confirmação de aceite, o que vem a finalizar o contrato. Caso Ag_2 não esteja satisfeito com a mesma, ele pode, ou encerrar a negociação ou enviar uma contraproposta, cujo procedimento para determina-la é similar ao do Ag_1, quando de sua proposta inicial. Trocas de propostas e contrapropostas seguem até que um das seguintes condições de término ocorram:

(i) um aceite por parte de Ag_1 ou Ag_2,

(ii) uma das partes decida por desistir do processo, ou

(iii) um evento externo, tal como, um valor que expresse um tempo limite.

O modelo "*Bazaar*" utiliza as redes de "*Bayesian*" [103], para actualizar o conhecimento e crenças que cada agente possui sobre o ambiente e sobre os demais agentes da comunidade. Retornando ao exemplo mencionado, e utilizando apenas um dos critérios de negociação, para efeito de simplificação, por exemplo o preço, do ponto de vista do Ag_2 (o agente consumidor) teremos:

(i) o Ag_2 conhece seus limites para o critério preço, porém não conhece os limites de Ag_1 (agente fornecedor) para este mesmo critério,

(ii) o Ag_2 é capaz de, baseado nas interacções com Ag_1, actualizar suas crenças (aprender) sobre o agente Ag_1. As crenças que o Ag_2 possui sobre o Ag_1 são representadas por um conjunto de hipóteses H_i,i=1,2,...,n. Por exemplo, relativo ao critério preço, H_1= \$100.00; H_2= \$130.00, etc., e cujas hipóteses possuem uma probabilidade estimada de $P(H_1)$=0.2, $P(H_2)$=0.35, etc. Uma actualização na rede ocorre quando, o agente consumidor (Ag_2) recebe nova informação,

ou proveniente de uma interacção com o agente fornecedor ou do ambiente. Esta nova informação, capacita o agente consumidor, obter novo conhecimento sobre os preços praticados pelo agente fornecedor, e
(iii) como resultado da aprendizagem, o agente consumidor espera adquirir melhor conhecimento sobre a estrutura de lucros do agente fornecedor, empregando-o no processo de negociação com o objectivo de obter vantagens.

O modelo "*Bazaar*" é um modelo de negociação entre agentes, com capacidades de, através da experiência, aprender sobre o domínio. Caracteriza-se por utilizar múltiplos critérios para o processo negocial, sendo o mesmo orientado pelo conhecimento que cada agente possui sobre os demais e sobre o ambiente, o qual foi aprendido durante as interacções.

3.3.2 Negociação para "*Rosenchein e Zlotkin*"

Em [104], Rosenchein e Zlotkin apresentam os seguintes componentes que determinam a coordenação entre agentes no domínio da negociação:

(i) o espaço de possíveis soluções. É o conjunto de possíveis acordos dentre os quais os agentes devem encontrar o entendimento. Este conjunto é chamado de conjunto de negociação ("*negotiation set*"),
(ii) um protocolo de negociação ("*negotiation protocol*"). Este protocolo define como podem, a partir do conjunto de negociação, os agentes chegarem a um entendimento. Os protocolos de negociação, restringem os possíveis cursos da negociação, através da especificações de acções e possíveis reacções, e
(iii) a estratégia de negociação ("*negotiation strategy*"). Dado o conjunto de negociação e o protocolo, a estratégia de negociação é responsável pelas decisões de um agente específico.

3.3.3 Negociação para "*J. P. Müller*"

Para Müller [105], o enquadramento do domínio da negocição na construção de uma arquitectura para agentes, apresenta aspectos que podem dificultar uma implementação pragmática, por exemplo: o modelo assume que o conjunto de negociação é totalmente determinado e conhecido por todos os participantes antes de iniciar a negociação.

Para contorna-los, Müller propõe que:

(i) o processo de negociação seja precedido por uma fase de coleta de informações para determinar se a negociação é requerida e possível, e para determinar o assunto da negociação,
(ii) o conjunto de negociação seja gerado, utilizando o seguintes modelo definido pela 7-upla:

$$NEG = (A, R, \#\#\#, N, U, P, S)$$

onde:

A é um conjunto de agentes, com seus respectivos estados mentais,
R é um conjunto de papéis, os quais os agentes podem desempenharem. Por exemplo: papel de líder ("*leader*"), de seguidor ("*follower*"), etc,
é a função que atribui os papéis aos agentes,
N é o conjunto de negociação,
U é a função utilidade para cada um dos agentes,
P é o protocolo de negociação dado pela dupla (K, ###). Dado um conjunto de tipos de mensagens (K), um protocolo é definido pelo conjunto das possíveis reacções (###) para cada um dos tipos de mensagens e para cada um dos papéis desempenhado pelos agentes. A semântica de uma mensagem, não depende apenas do tipo de mensagem de "*per se*", mas também do seu conteúdo. Por exemplo, considerando a função do protocolo, "###" pode restringir as possíveis reacções para uma mensagem "PROPOSE p, p ### N", como sendo ou "ACCEPT" ou "REJECT". A decisão, qual de ambas é apropriada, depende fundamentalmente da atitude do agente em relação ao conteúdo de "p", a qual por sua vez pode depender do estado mental do agente i.e. suas crenças, e
S é um conjunto de estratégias de negociação. Enquanto o protocolo de negociação, restringe os possíveis cursos da negociação, eles não providenciam as descrições das possíveis negociações. Em cada passo de um processo de negociação, um agente deve escolher entre diferentes alternativas. As estratégias são definidas em relação ao protocolo e aos papéis desempenhados pelos agentes. A decisão de um agente em uma situação específica e em um protocolo, depende não apenas da prévia acção dos parceiros da negociação, mas também de informações adicionais sobre o estado actual do processode negociação,

(iii) o conhecimento comum pode ser alcançado, determinado um agente para calcular o conjunto de negociação e transmiti-lo para os demais,
(iv) os agentes envolvidos devem concordar com o protocolo de negociação e então os papéis lhes são atribuídos,
(v) se houver simetria nos protocolos, i.e., o papel atribuído não é restringido pela natureza do protocolo, os agentes devem concordar com o papel designado. Para este caso, é assumido que os agentes "lançam a moeda" para determinarem que papéis desempenharão no processo negocial. Em um protocolo assimétrico, a atribuição do papel é determinada "*apriori*". Um exemplo de um protocolo assimétrico é o da rede de contrato ("*contract net protocol*")[76] onde, para um dado assunto, é bem definido quem será o gestor e quem serão os agentes que farão as ofertas, e
(vi) finalmente, cada agente escolhe localmente sua estratégia e a negociação inicia.

O protocolo utilizado neste modelo é similar ao protocolo de concessão monótono ("*monotonic concession protocol*")[104]. Cada agente inicia com a oferta que maximiza suas utilidades (U). Se o agente envolvido nesta negociação rejeita uma oferta, ele propõe uma contra-oferta. No caso da oferta ser rejeitada, o agente que a fez, retira o correspondente elemento do conjunto de negociação (N). De maneira

análoga, se um agente rejeita uma oferta, ele exclue este elemento do seu conjunto de negociação.

O encerramento de um processo de negociação, é dado pelo seguinte teorema:

> Faça **N** ser um conjunto finito de negociação, e faça u_i ### U induzir a total ordenação em **N**. Então, a negociação a qual é implementada utilizando o modelo, encerra depois de um finito número de execuções.

3.3.4 Sequenciamento da Produção como um Processo de Negociação

Encontramos em [106] o processo de negociação visto como o sequenciamento da produção. O problema de sequenciamento da produção envolve uma fábrica, gerando um produto tal que, quando processado, a sequência satisfará a solicitação e minimizará os custos totais. O sequenciamento da produção pode ser considerada como um problema de negociação, na qual as células de manufacturas da fábrica negociam sobre a sequência de fabrico do produto, de modo a favorecer a distribuição dos custos.

A ideia básica desta aproximação, apoia-se em:

(i) uma fábrica é constituída de várias células de manufactura,

(ii) a cada célula está associada um custo, e

(iii) estes custos implicitamente definem a preferência pela sequência de produção de um produto.

Em um ambiente multiagente, cada célula é um agente, os quais estão envolvidos na negociação de uma sequência de produção. Cada agente comporta-se de forma a melhorar sua utilidade através da redução dos custos. Um ambiente multiagente para modelar este problema apresenta as seguintes vantagens:

(i) modularidade. O problema de sequenciamento da produção é um problema de optimização. Se pensarmos em adoptar modelos matemáticos, empregando técnicas de optimização, a modelação pode tornar-se não realista, devido a sua complexidade e a actual não disponibilidade dos mesmos. Porém dividí-lo em subproblemas de tamanhos possíveis de gestão e abordá-los através de metodologias de sistemas multiagente, torna-se uma solução mais realista na gestão da complexidade,

(ii) modelação adequada. Soluções monolíticas e centralizadas, não representariam uma estrutura organizacional característica de uma fábrica,

(iii) flexibilidade. Sendo uma fabrica, uma organização não estática, isto é apresenta constantes alterações em termos de equipamentos, técnicas de produção, etc., a aproximação multiagente, na qual os custos são distribuídos pelos agentes, possibilita que estas alterações sejam facilmente assimiladas, e

(iv) paralelismo, o qual contribui para na optimização do tempo de busca da solução.

O processo de encontrar uma sequência de produção para um dado produto, pode ser visto como um problema de negociação, no qual cada agente que representa uma célula de manufactura, negocia sobre sua sequência de produção, em busca de

minimizar seus custos locais e reduzir os custos globais de fabrico do bem. Há dois tipos de propostas que os agentes podem fazer durante o processo de negociação:

(i) os agentes podem negociar sobre produtos individuais para, de maneira incremental, desenvolverem uma sequência óptima de produção, e

(ii) os agentes negociarem sobre a completa sequência de produção, para encontrarem uma que minimize os custos.

Esta aproximação considera apenas o segundo tipo de proposta, devido a mesma apresentar uma proximidade com o mundo real. Tipicamente, no mundo real, o processo inicia com a apresentação de uma completa sequência de produção que se aproxime do óptimo, seguida de sucessivos refinamentos. Neste caso os agentes negociam sobre a completa sequência de produção. Para o modelo, algumas das características de um processo de negociação tratado pela teoria dos jogos não são relevantes. Como os agentes envolvidos são modelados para encontrarem o objectivo comum de redução do custo total de fabrico, a busca pelo equilíbrio de "*Nash*", por exemplo é irrelevante, uma vez que o mesmo assume que cada participante apresenta um comportamento de maximizar suas utilidades individuais. É também assumido que os agentes são benevolentes, não havendo propostas enganosas, pois o objectivo de maximização de suas próprias utilidades é desviado para a utilidade da organização.

O mecanismo de negociação utilizado é uma extensão e adaptação do protocolo monótono de concessão ("*monotonic concession protocol*") [104]. A negociação é executada em sessões sucessivas ("*rounds*"), sendo que na primeira, cada agente toma uma parte activa, propondo algum negócio. Se um negócio proposto satisfaz cada um dos agentes envolvidos, então a negociação encerra de forma satisfatória. Caso contrário, a negociação passa para uma próxima sessão, na qual um subconjunto dos agentes activos devem conceder. Para as concessões, existem três possibilidades:

(i) o agente é capaz de propor um negócio, o qual represente uma verdadeira concessão,

(ii) o agente é incapaz de fazer uma verdadeira concessão, mas é capaz de fazer outra proposta, e

(iii) o agente exauriu suas possibilidades de propostas, e então retira-se da negociação.

Os agentes que não concedem em alguma das sessões, desloca o mesmo negócio para uma próxima sessão. Desta forma, a negociação continua com agentes concedendo e retirando, até que finalmente eles possam encontrar um negócio no qual eles concordam. Os agentes envolvidos no processo de negociação pesquisam o espaço de negociação, na tentativa de encontrar um negócio mutualmente aceitável.

Em relação ao algoritmo, três questões devem ser respondidas:

(i) a escolha, por parte de um agente, da primeira proposta,

(ii) quem deve conceder, em uma dada sessão, e

(iii) se um agente faz uma concessão, em quanto ele deve conceder.

Em relação a escolha da primeira proposta por parte de um agente, quando a negociação inicia, ele deve seleccionar algum negócio para propor, a partir de um conjunto de negociação. A escolha recai no negócio pertencente ao conjunto de negociação que:

(i) ainda não tenha sido proposto (que não esteja na mesa de negociação),

(ii) que apresente o maior lucro dentre eles, e
(iii) que minimize os custos de fabrico.

O negócio escolhido representará a melhor proposta que um agente ainda é capaz de fazer.

Quanto a questão de quem deve conceder, esta cabe ao agente que tiver menos a perder com a concessão, i.e. para o qual a decisão de conceder representa o menor risco, o qual é dado por:

$$\text{risco de um agente em uma dada sessão} = \frac{\text{utilidade desse agente perder por ter concedido}}{\text{utilidade desse agente perder por não ter concedido}}$$

E, relativo ao quanto deve ser concedido por um agente, para o caso de concessão, i.e. como deveria ser a próxima proposta, as seguintes propriedades devem ser consideradas:

(i) que a proposta não tenha sido feita em sessões anteriores de uma negociação,
(ii) que a proposta seja ao menos satisfatória para todos os agentes, quanto a sua proposta corrente e melhor para pelo menos um outro agente,
(iii) que devido a proposta, se possa obter uma melhora (tempo, qualidade, etc.) no conjunto de toda a produção,
(iv) o agente deve conceder o suficiente para que numa próxima sessão, este agente não necessite conceder, e
(v) a proposta feita por um agente será a mínima concessão que este pode fazer.

Estas propriedades representam as heurísticas que orientam a busca da solução no espaço de negociação. Pode ocorrer que um agente concessionário seja incapaz de encontrar uma proposta que satisfaça estas propriedades. Neste caso, o agente deve recalcular suas propostas contidas no espaço de negociação. Se o espaço de negociação estiver vazio, o agente retira-se da negociação.

3.4 O Processo de Selecção e de Contratação de Colaboradores em "*ARTOR*"

O processo de selecção e de contratação de colaboradores, em uma sociedade de organizações, possui o seguinte enunciado:

(i) dado um conjunto de organizações, as quais buscam a cooperação para o provimento de bens/serviços,
(ii) dado um conjunto de organizações, as quais tem como um dos seus objectivos cooperarem no provimento de bens/serviços,
(iii) dado um conjunto de critérios o qual expressa as características desejadas dos bens/serviços e da organização que irá provê-los,
(iv) as organizações que buscam a cooperação, após uma quantidade de interacções com as interessadas em cooperarem, devem ser

capazes de seleccionar uma ou caso contrário, os critérios devem ser replaneados, e

(v) as organizações envolvidas nesta relação, deverão ser capazes de aprender com as interacções, a princípio, no domínio da mesma.

Para o sistema "*ARTOR*", o processo acima enunciado [107] apresenta as seguintes características:

(i) é uma das relações do conjunto possível das relações sociais entre agentes envolventes (organizações),

(ii) ocorre através de **agentes executores** dos **agentes envolventes**, os quais respectivamente desempenham; os papéis de **organizadores** ou **respondentes**, dependendo dos objectivos de: buscar cooperação ou de estar interessado em cooperar,

(iii) utiliza um processo de selecção baseado em critérios, através dos quais um organizador escolherá um dos respondentes,

(iv) caso não tenha havido interacções passadas, um agente não possui prévio conhecimento sobre os demais, i.e. o modelo dos agentes conhecidos (***Am***) da base de conhecimento corporativa (***Ck***) é igual {}, e

(v) as competências dos agentes executores, suportadas pelas suas capacidades e conhecimentos, apresentam os seguintes comportamentos [fig. 64] para a relação de selecção e contratação de colaboradores:

Organizador	**Respondente**
Comportamentos: . convidar a sociedade a participar, . avaliar as contrapropostas, . detectar conflitos, . negociar os valores dos critérios, . efectivar a contratação, e . aprender no domínio da negociação.	**Comportamentos:** . decidir sobre o convite, . avaliar as propostas, . negociar os valores dos critérios, e . aprender no domínio da negociação.

[fig. 64] Comportamentos para a relação social de selecção e de contratação de colaboradores

O processo de selecção e de contratação de colaboradores é baseado em uma **lista de critérios** (***CL***), para cada bem/serviço a ser provido. Uma lista de critérios, é formada pela conjunção de unidades denominadas de **critérios de selecção** (***SC***).

Os critérios de selecção:

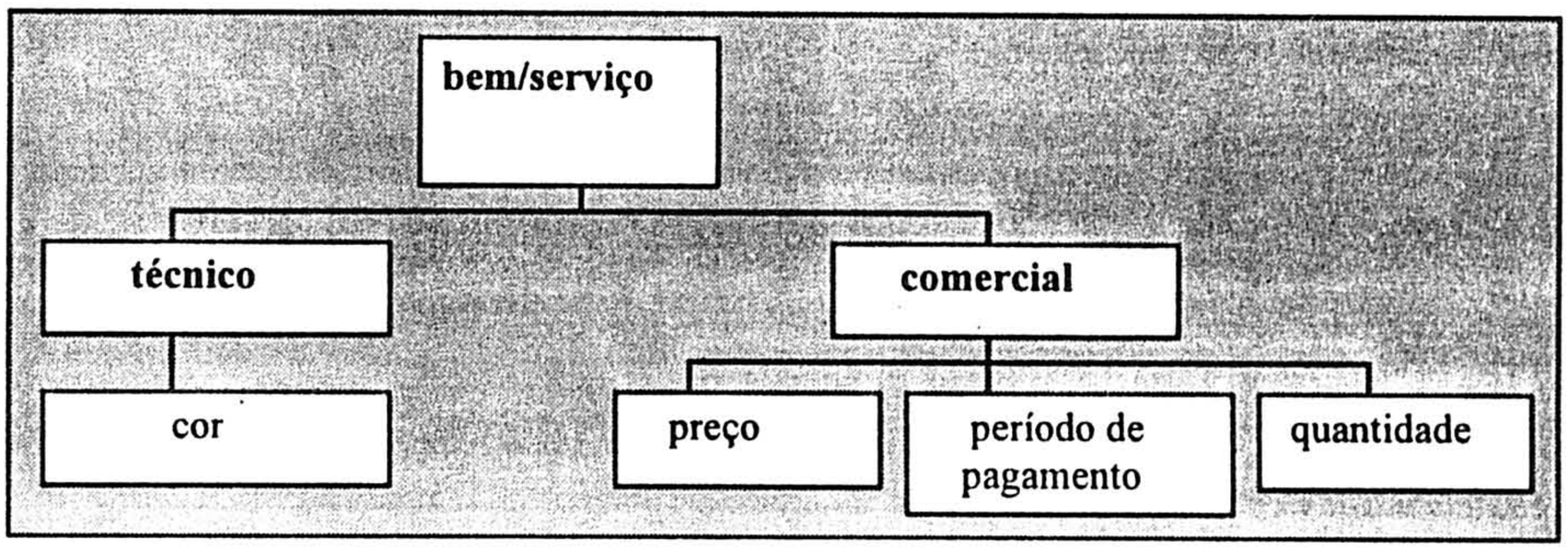
(i) expressam quais as dimensões, que o bem/serviço e a organização, potencial provedora, serão avaliadas, e
(ii) apoiam, o organizador, na busca pela especificação e pela qualidade (satisfação) do bem/serviço a ser provido pelos respondentes.

A estrutura em árvore foi adoptada para representar a taxinomia das dimensões a serem avaliadas. Cada uma das folhas da árvore representa um critério de selecção (*SC*).

```
/*************************************************************/
dimensoes_avalicao( [
        [ Dimensao, [Atributos_dimensão]],
        ...,
        [ Dimensao, [Atributos_dimensão]],
          ]).
/*************************************************************/
```

No exemplo, a seguir [fig. 65], as dimensões de avaliação, do produto/serviço a ser provido, são:

(i) técnica, contendo como atributo a cor, e
(ii) comercial, contendo como atributos o preço, o período de pagamento e a quantidade.



[fig. 65] Taxinomia dos domínios de avaliação

Cada critério de selecção (*SC*) é uma 5-upla:

$$sc_i(Id_i,\ Vd_i,\ Va_i,\ Pr_i,\ Sm_i)$$

onde:

Id_i é a identificação do critério,
Vd_i é o valor, do critério, que **satisfaz** o agente,
Va_i é representado pela dupla (Vac_i, Fed_i), onde:, e
Vac_i representa o domínio dos valores válidos de sc_i, e
Fed_i indica a medida de satisfação, em relação ao valor que satisfaz, dos **demais valores do domínio** [fig. 66]:
right quando uma maior satisfação (satisfaz mais) encontra-se nos valores situados a direita do valor de referência (Vd_i) que satisfaz o critério. Neste caso, se existir, a

menor satisfação (satisfaz menos) está nos valores localizados a esquerda,

left quando uma maior satisfação (satisfaz mais) encontra-se nos valores situados a esquerda do valor de referência (Vd_i) que satisfaz o critério. Neste caso, se existir, a menor satisfação (satisfaz menos) está nos valores localizados a direita,

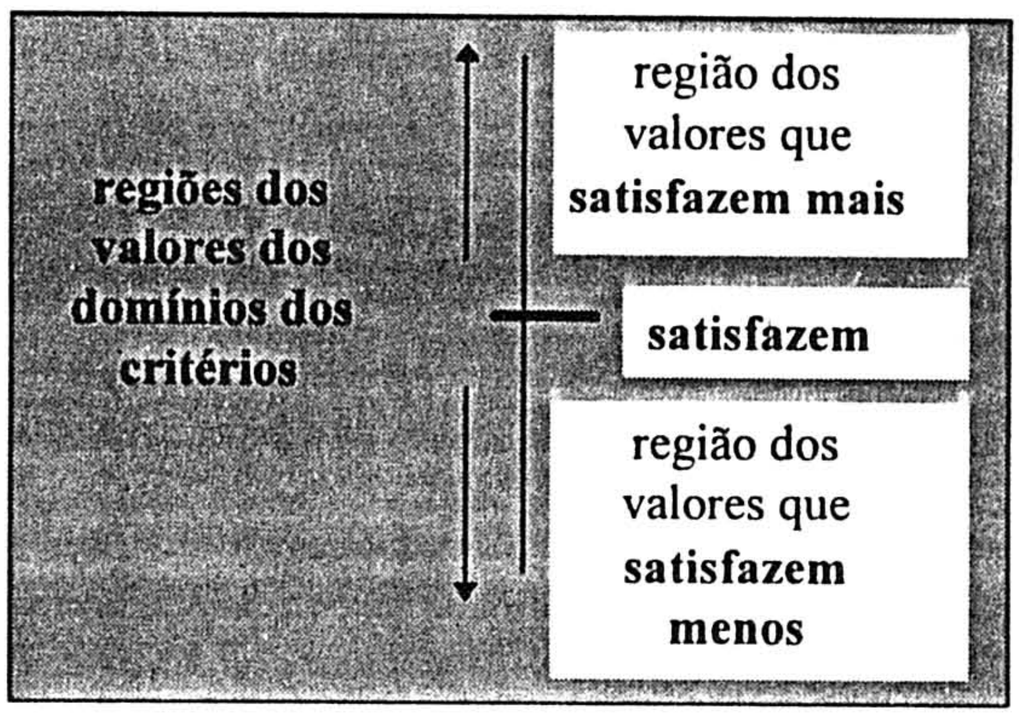
none quando não há preferência em relação aos demais valores do domínio (Vac_i),

Pr_i é o valor da **utilidade** da sc_i para o agente (Pr_i ### Cardinais),

Sm_i representa o estado do critério relativo a instanciação do valor:

grounded a primeira proposta da sc_i será feita especificando um valor, e

free a primeira proposta da sc_i será feita sem especificar um valor



[fig. 66] Regiões dos valores dos domínios dos critérios

Uma lista de critérios (*CL*) é definida por:

$$CL_{produto/serviço}(sc_1, sc_2, \ldots, sc_i, \ldots, sc_n)$$

O produto Cartesiano (*D*) dos domínios de cada um dos critério de selecção (*SC*), pertencentes a uma lista de critérios (*CL*), define um **espaço de possibilidades** (*PS*) de critérios de selecção.

$$PS \in CL = D_{SC1} \times D_{SC2} \times \ldots D_{SCi} \times \ldots \times D_{SCn}$$

Um espaço de possibilidades de uma "*CL*" contém os possíveis, para um agente e para um específico processo de selecção, critérios de avaliação e seus respectivos valores com os quais pode solicitar ou oferecer cooperação.

Para identificar a utilidade, para um agente, relativa aos critérios de selecção (*SC*) de uma lista de critérios (*CL*), dois diferentes parâmetros de utilidade estão presentes:

(i) o primeiro parâmetro de utilidade, denotado por "Pr_i", representa a importância do critério, em relação aos demais, em

uma específica lista de critérios (*CL*). Esta utilidade é representada por cardinais, podendo ser o valor de "Pr_i" de um critério, maior, menor ou igual ao "Pr_k" de um outro. Apresentando como exemplo os critérios cor e período de pagamento,

```
[cor, azul, [[azul, preta], none], 1, ground],...
[período_de_pagamento, 30, [[vista, 30, 60, 90, 120],
right], 3, ground],...
```

o critério cor ("Pr_i" = 1) é de menor utilidade em relação ao critério prazo de pagamento ("Pr_k" = 3), para o agente em uma específica lista de critérios (*CL*),

e

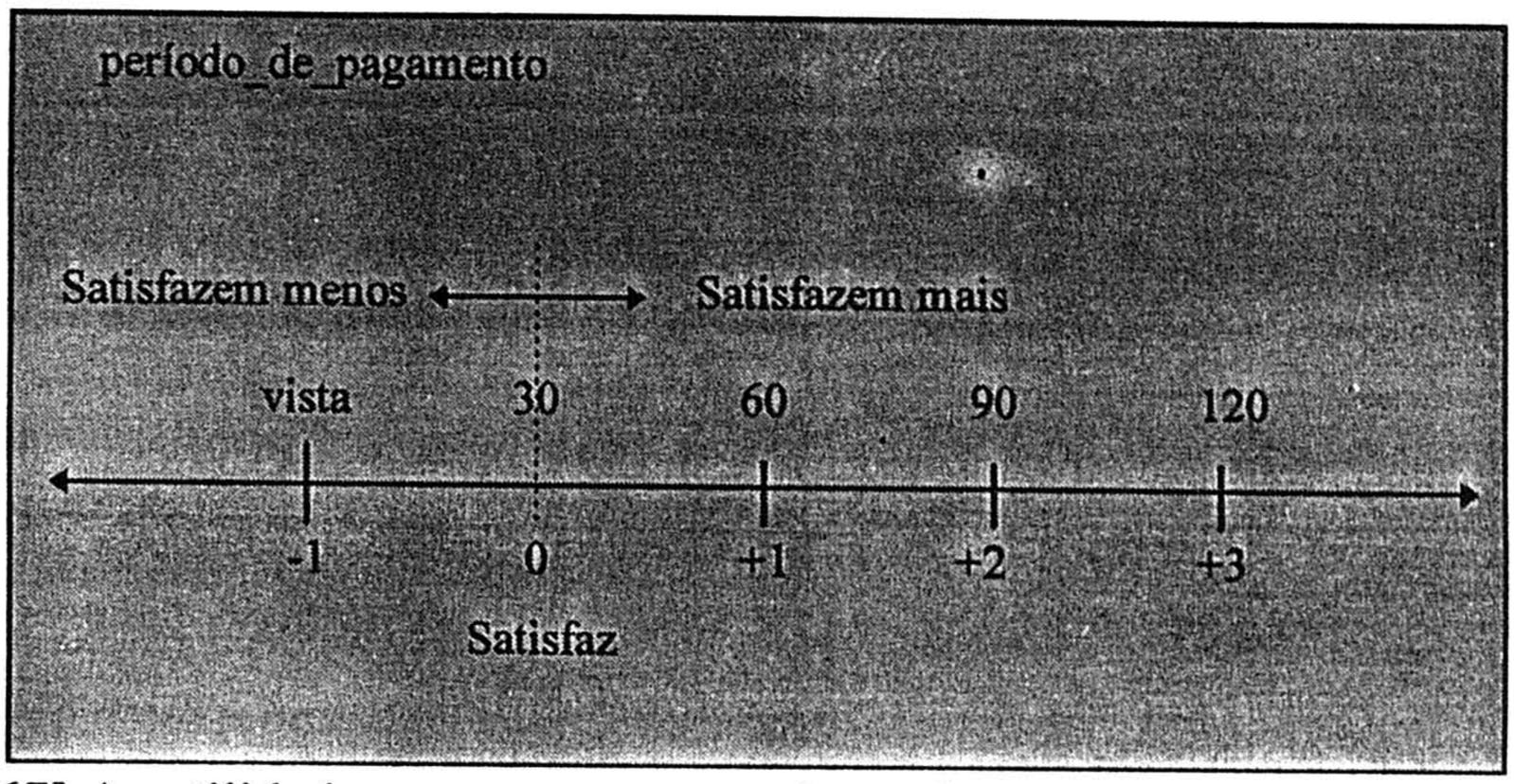
(ii) o segundo parâmetro de utilidade, denotado por "Fed_i", expressa em qual sentido, os valores do domínio, satisfazem mais, ou são indiferentes (left, right, none), em relação ao valor(es) que satisfaz(em). O sentido contrário, são os valores que satisfazem menos.

Dado o domínio de um critério de selecção e assumindo zero (0) para o(s) valor(es) que satisfaz(em), os demais podem satisfazer mais ou menos em relação a este, diferenciados de uma unidade. Os valores que satisfazem mais são positivos (+) e os que satisfazem menos são negativos (-).

Como exemplo, tomemos o critério de avaliação:

```
..., [período_de_pagamento, 30, [[vista, 30, 60, 90, 120],
right], 3, ground], ...
```

Neste exemplo, existem cinco valores no domínio (vista, 30, 60, 90, 120), com os quais um agente, no critério período de pagamento, pode solicitar ou prestar cooperação. Representando os valores do domínio em uma recta orientada [fig. 67], fixando como origem o valor que satisfaz (30) e variando de uma unidade positiva ou negativa os demais valores, dependendo da satisfação ser maior ou menor, teremos:



[fig. 67] As utilidades, para um agente, dos valores do domínio de uma "*SC*"

A utilidade de uma instância válida de um critério de selecção (*CS*), em uma lista de critérios (*CL*), é dada por:

$$\text{utilidade_instância} = (Pr_i \times \text{valor relativo da instância})$$

Utilizando como exemplo o critério período de pagamento com o valor noventa (90), a utilidade da instância é:

$$\text{utilidade_instância} = 3 \times (+2) = 6$$

Para ilustrar o conteúdo que se segue, sobre o processo de negociação em "*ARTOR*", doravante adoptaremos o cenário (apresentado nas experiências citadas no capítulo II dessa dissertação) composto por três agentes envolventes (organizações):

(i) " ***artor_org_A***", agente envolvente no papel de organizador,
(ii) " ***artor_org_1***", agente envolvente no papel de potencial respondente, e
(iii) " ***artor_org_2***", agente envolvente no papel de potencial respondente,

e seus respectivos agentes executores, responsáveis pelo processo de selecção e de contratação de colaboradores, em cada uma das organizações:

(i) agente executor "***artor_org_A_selection_expert***" de "***artor_org_A***",
(ii) agente executor "***artor_org_1_supply_expert***" de " ***artor_org_1***", e
(iii) agente executor "***artor_org_2_supply_expert***" de " ***artor_org_2***".

O organizador, através de seu agente "***artor_org_A_selection_expert***", procura na sociedade de organizações, parceiros que colaborem no desenvolvimento de um produto (P_1). Esta procura ocorre na forma de um processo de selecção, através do qual um dos potenciais respondentes, "***artor_org_1_supply_expert***"ou " ***artor_org_1_supply_expert***", será o contratado.

A representação, na estrutura de uma "*CL*", da instância utilizada pelo organizador " ***artor_org_A***", para a contratação de um respondente, para o provimento de um produto P_1 nos domínios de avaliação:

(i)técnico:

(i.i) cor, e

(ii) comercial:

(ii.i) preço,

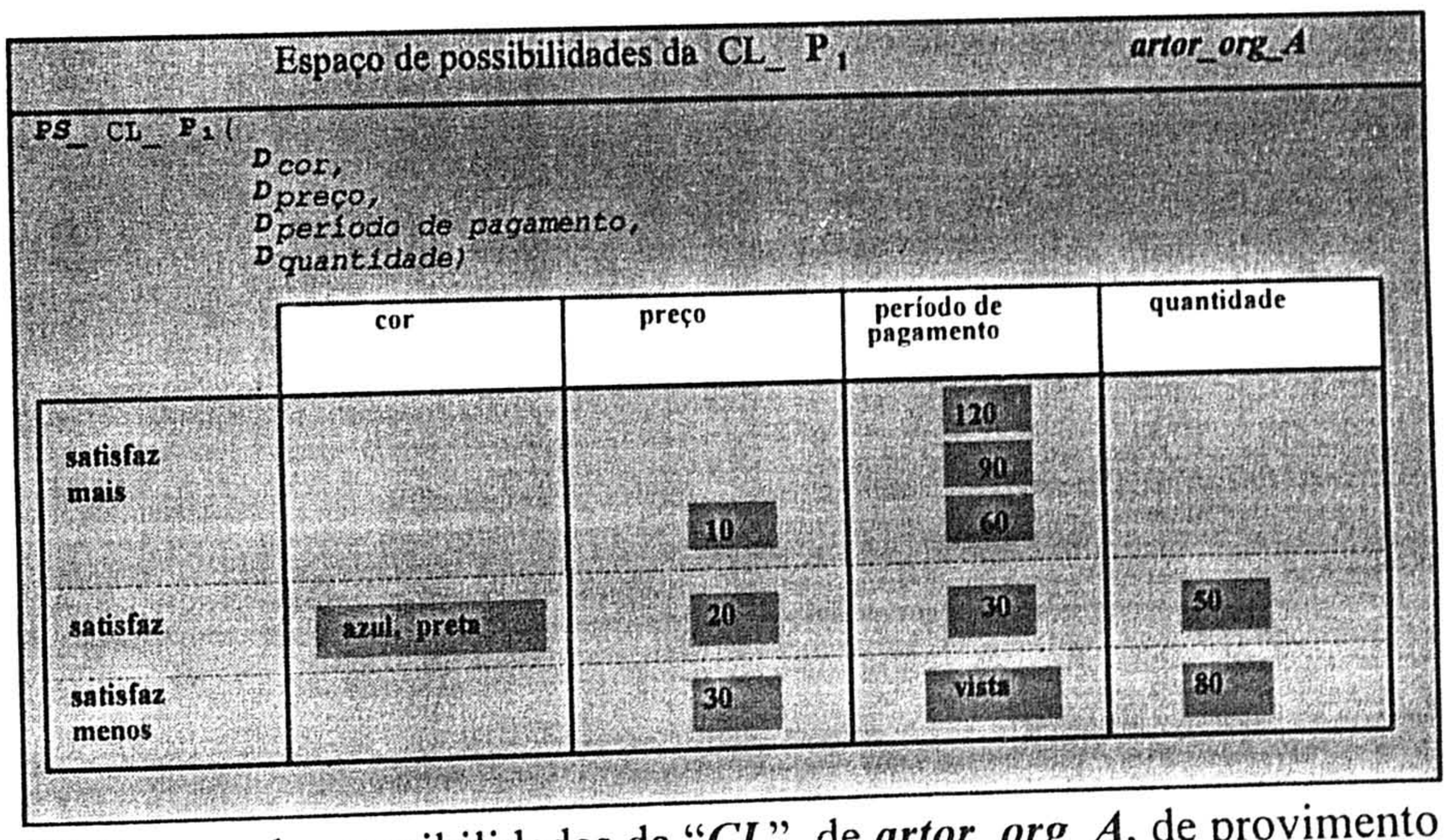
(ii.ii) período de pagamento, e

(ii.iii) quantidade,

é:

```
CL_P1([
[cor, azul, [[azul, preta], none], 1, ground],
[preço, 20, [[10,20,30], left], 3, ground]
[período_de_pagamento, 30, [[vista, 30, 60, 90, 120],
right], 3, ground],
[quantidade, 50, [[50, 80], left], 2, ground]).
```

A representação do correspondente espaço de possibilidades (*PS*) da CL_P_1 é apresentada a seguir [fig. 68].



[fig. 68] Espaço de possibilidades da "*CL*", de *artor_org_A*, de provimento de P_1

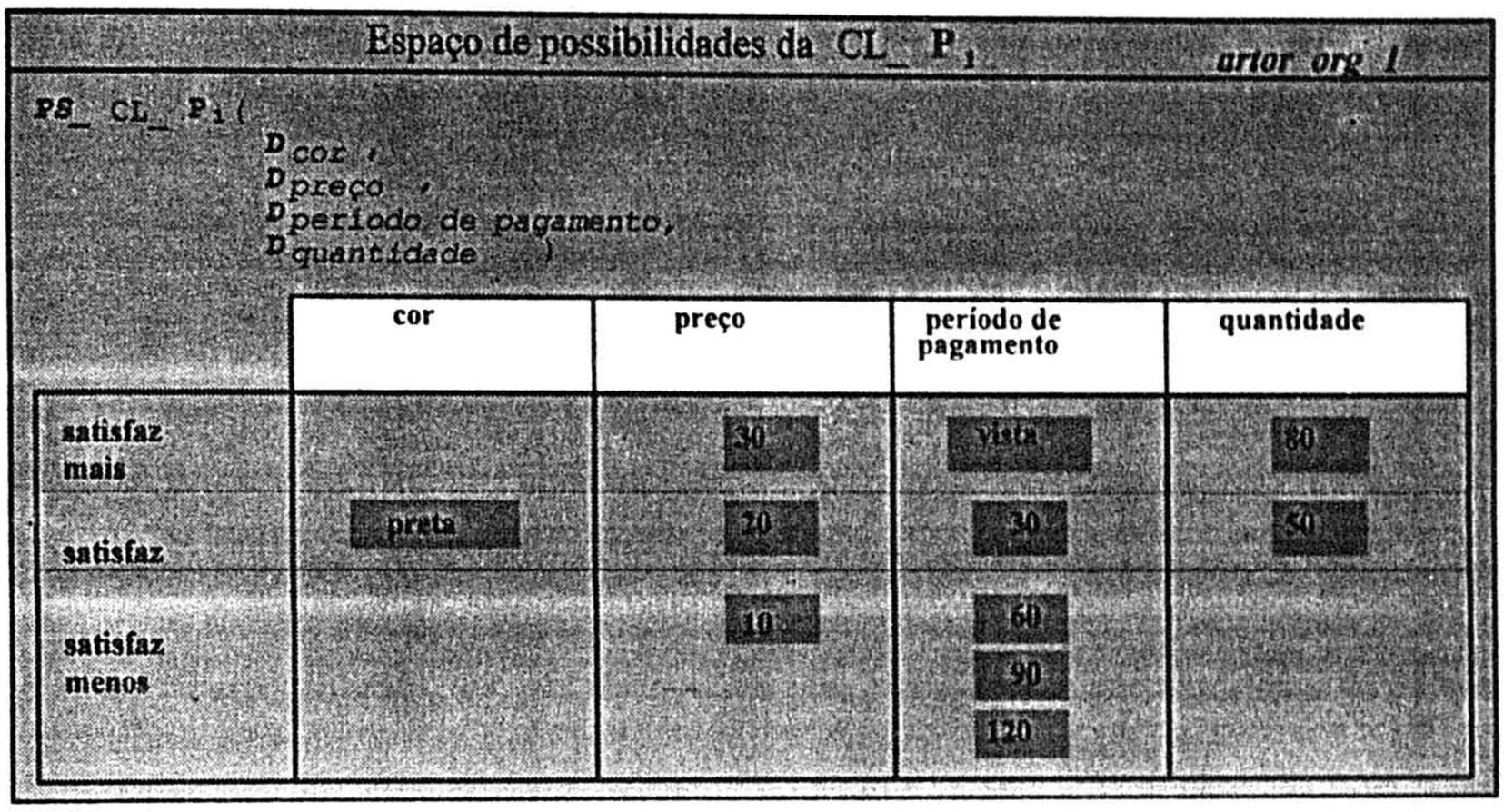
Para os demais componentes da sociedade de organizações, as "*CLs*" são as seguintes:

(i) para " *artor_org_1*":

```
CL_P1([
```

```
[cor, preta, [[preta], none], 2, ground],
[preço, 20, [[10,20,30], right], 1, ground]
[período_de_pagamento, 30, [[vista, 30, 60, 90, 120],
left], 3, ground],
[quantidade, 50, [[50, 80], right], 3, ground]).
```

a representação do correspondente espaço de possibilidades (*PS*) da CL_P_1 é apresentada a seguir [fig. 69], e



[fig. 69] Espaço de possibilidades da "*CL*", de "*artor_org_1*", de provimento de P_1

(ii) para " *artor_org_2*":

```
CL_P1([
[cor, azul, [[azul, preta], none], 2, ground],
[preço, 20, [[20, 30], right], 3, ground]
[período_de_pagamento, 60, [[vista, 30, 60, 90, 120],
left], 1, ground],
[quantidade, 50, [[50], none], 1, ground]).
```

a representação do correspondente espaço de possibilidades (*PS*) da CL_P_1 é apresentada a seguir [fig. 70].

Espaço de possibilidades da CL_P_1 — *artor_org_2*

```
PS_CL_P1(
        D cor ,
        D preço ,
        D período de pagamento,
        D quantidade    )
```

	cor	preço	período de pagamento	quantidade
satisfaz mais		30	vista 30	
satisfaz	azul, preta	20	60	50
satisfaz menos			90 120	

[fig. 70] Espaço de possibilidades da "*CL*", de "*artor_org_2*", de provimento de P_1

3.4.1 O Convite à Sociedade de Organizações

O convite à sociedade é um evento disparado pelo organizador. Tem por objectivo levar o processo de selecção e de contratação de colaboradores ao conhecimento da sociedade e aguardar que os agentes envolventes interessados no mesmo se manifestem.

Para tornar disponível o convite, o organizador utiliza o quiosque "*News_Stand*" (Capítulo II desta dissertação), para afixação do mesmo.

Em referência ao cenário mencionado, o agente "***artor_org_A_selection_expert***" envia a mensagem ao quiosque "*News_Stand*" de pedido de afixação de notícias no seguinte formato:

```
/*****************************************************/
news_offer( news_id( News_id),
          news_subject( News_subject),
          news_content( News_content),
          news_deadline( News_deadline),
          news_boundary( Boundary_type, Boundary_value)
          ).
```

onde:

"*News_id*" é a identificação da notícia à ser afixada,

"*News_subject*" é o título da notícia,

"*News_content*" é o conteúdo da notícia. Para o processo de selecção e de contratação

é:

<List_of_constraints, Qty>

onde:

List_of_constraints é a lista de critérios a serem utilizados no processo de selecção, contendo:

[Id, Vd, Negotiation, ...]

onde:

Id é a identificação do critério,

Vd é o valor inicial do critério, e

Negotiation se o critério:

permite a negociação = y, ou

não permite a negociação = n,

Qty é a quantidade máxima de iteracções,

quando pelo menos um critério permite a negociação.

"*News_deadline*" é a data limite para resposta, por parte dos agentes nela interessados e caso a notícia requeira uma data de validade. Ex: promoção válida até...

"*news_boundary*" é o limite para veiculação da notícia i.e. se a notícia estará disponível até uma certa data e/ou em uma certa quantidade:

"*Boundary_type*" = {data, quantidade}

"*Boundary_value*" é o valor correspondente ao tipo.

```
news_offer( news_id(seleccao_contratacao),
  news_subject(desenvolvimento_de_produto_pl),
  news_content ([cor, azul, y,
                 preco, 10, y,
                 período_de_pagamento, 120, y,
                 quantidade, 50, y
                ],
                 6
               ),
  news_deadline('970202231155' ),
  news_boundary( quantity, 2)
  ).
```

e cuja a mensagem resposta, do quiosque, tem o seguinte formato:

```
comfirming_news_offer( news_id( News_id),
        news_subject( News_subject),
        news_acknowledgment( News_acknowledgment)
        ).
```

onde:

News_acknowledgment = {accepted, not_accepted}

```
comfirming_news_offer( news_id(seleccao_contratacao),
        news_subject(desenvolvimento_de_produto_pl),
        news_acknowledgment(accepted)     ).
```

```
/*************************************************************/
```

Os valores contidos na lista de critérios enviada pelo organizador ao quiosque, reflecte a estratégia utilizada pelo agente, no processo de selecção e de contratação de colaboradores. A apresentação desses valores e a aplicação da estratégia para gerá-los serão tratados na próxima secção " A Avaliação e a Situação de Conflito".

Os demais agentes da sociedade, "*artor_org_1_supply_expert*" e "*artor_org_2_supply_expert*", caracterizam-se por apresentarem um comportamento de procura constante de negócios, nos seus domínios de actuação. Este comportamento orienta-os a contactar o quiosque "*News_Stand*" em busca das oportunidades, através da mensagem no seguinte formato:

(i) de solicitação de notícias:

```
/*****************************************************************/

asking_for_news( news_subject( News_subject)
                 ).
```

```
asking_for_news( news_subject (desenvolvimento_de_produto_p1)
                 ).
```

```
/*****************************************************************/
```

(ii) e cuja resposta é:

```
/*****************************************************************/

(i)answering_ask_for_news( news_id( News_id),
        news_content( News_content),
        news_deadline( News_deadline),
        news_from_agent( Agent_name, Hostname, Portnumber)
        ).
```

onde:

"*news_from_agent*" é a identificação do agente que disponibilizou a notícia

```
(ii)answering_ask_for_news(
        that_is_all( News_subject)
        ).
```

onde:

"*that_is_all*" informação de final de envio de notícias solicitadas

```
answering_ask_for_news( news_id(seleccao_contratacao),
        news_content ( [cor, azul, y,
                        preco, 10, y,
                        período_de_pagamento, 120, y,
                        quantidade, 50, y
                        ],
                        6
                      ),
        news_deadline('970202231155'),
        news_from_agent(artor_org_A_selection_expert,
                        turing, 3620)

        ).
answering_ask_for_news(
        that_is_all(desenvolvimento_de_produto_p1)
        ).
```

```
/**************************************************************/
```

3.4.2 A Avaliação das Propostas e a Situação de Conflito

Após afixar no quiosque o convite e caso haja respondentes, o organizador passará a estabelecer uma comunicação ponto à ponto com os mesmos. Não havendo respondentes, até o limite contido no "*news_deadline*", o processo de selecção, da parte do organizador, é encerrado.

Quando são recebidas as mensagens dos respondentes, de aceitação ao convite, o organizador passa a executar a avaliação, através de sua capacidade de "especialidade" (Capítulo II dessa dissertação).

A avaliação inicial, das mensagens recebidas ao convite, está baseada na comparação entre:

(i) a lista de critérios afixada no quiosque, e

(ii) as listas recebidas.

```
/**************************************************************/
answer_offer( news_id( News_id),
              news_subject( News_subject),
              answer_content( Answer_content)
              ).
```

onde:

"*News_id*" é a identificação da notícia afixada,

"*News_subject*" é o título da notícia,

"*Answer_content*" contém a resposta ao convite:

<Message_number, Offer>

onde:

Message_number: é um cardinal, iniciando com zero (resposta ao convite), que indica a sequência das mensagens enviadas, para o processo de selecção, *e*

Offer: contém a lista critérios com seus valores de resposta:

[Id,Vr, ...]

onde:

Id é a identificação do critério, e

Vr é o valor do critério.

```
/**************************************************************/
```

Passamos a denominar de **oferta inicial** a lista de critérios inicial enviada pelo organizador, e de **contra-oferta iniciais** as listas de critérios geradas pelos respondentes.

A geração da oferta inicial, pelo organizador, e das contra-ofertas iniciais, pelos respondentes, são baseadas no estilo do agente envolvente, em estratégias e em tácticas. Cada um dos agentes, participantes da sociedade de organizações possui os seguintes conhecimentos:

(i) estilo do agente. Em um processo de tomada de decisão, o conhecimento acerca do perfil do agente, actua como orientador do

mesmo. Em um processo de selecção e de contratação de colaboradores, o estilo do agente orientará qual a estratégia a ser utilizada. Se o agente for do estilo "*win_win*", considerará as contra-ofertas do potencial parceiro, relaxando alguns dos critérios utilizados para negociar. Caso contrário, se o perfil for "win_lose" utilizará sua posição no mercado, e tenderá a não relaxar, ou relaxar muito pouco os critérios. O conhecimento sobre o estilo do agente se encontra na base de conhecimento corporativa ("*Ckb*") do agente envolvente:

```
/*************************************************************/
i_am( artor_org_A_corporate_knowledge_data_base).
/*************************************************************/
% Self Model - modelo do auto conhecimento
/*************************************************************/
% Self Model - Sobre o perfil do agente
/*************************************************************/
...
skill( style, artor_org_A, win_win).
...
/*************************************************************/
```

(ii) as estratégias e tácticas são conhecimentos que se encontram na base de conhecimento individual ("*ikb*") de agentes executores (pertencentes a agentes envolventes), dotados de perícia para participar de um processo de selecção e de contratação de colaboradores. Estes conhecimentos são representados na estrutura de "*frames*", no seguinte formato:

```
/*************************************************************/
frame (Agent_style, [
  strategy, [ Strategy_name],
  tactics, [ Tactics_name],
  oper, [Oper_names]]).
```

onde:

" *Agent_style* " é o perfil do agente envolvente,

"*Strategy_name*" contém o nome da estratégia,

"*Tactics_name*" contém o nome do predicado a ser chamado, o qual implementa a táctica correspondente a estratégia, e

"*Oper_names*" contém o nome dos operadores para a implementação da táctica.

```
/*************************************************************/
```

Em referência ao cenário, o organizador recebe as seguintes mensagens dos respondentes:

(i) "*artor_org_1_supply_expert*":

```
answer_offer( news_id(seleccao_contratacao),
  news_subject(desenvolvimento_de_produto_p1),
  answer_content(0,
                 [cor, preta,
```

```
                preco, 10,
                período_de_pagamento, vista,
                quantidade, 80])
            ).
```

(i) "*artor_org_2_supply_expert*":

```
answer_offer( news_id(seleccao_contratacao),
  news_subject(desenvolvimento_de_produto_p1),
  answer_content(0,
                [cor, azul,
                preco, 30,
                período_de_pagamento, 120,
                quantidade, 50])
            ).
```

A avaliação, pela comparação individual [fig. 71] de cada um dos critérios, para cada um dos respondentes, em relação aos valores propostos pelo organizador, apresentam diferenças devido a discordância de valores. Esta situação se apresenta como uma situação de conflito a qual utiliza a metodologia de negociação para a resolução dos conflitos.

	oferta inicial	contra oferta inicial	
	"*artor_org_A_selection_expert*"	"*artor_org_1_supply_expert*"	"*artor_org_2_supply_expert*"
cor	azul	***preta***	azul
preço	10	10	***30***
período_de_pagamento	120	***vista***	120
quantidade	50	***80***	50

[fig. 71] Comparação dos valores dos critérios ofertados e contra-ofertados

Em relação as gerações da oferta inicial e das contra-ofertas iniciais, que levaram à estas diferenças, o organizador e os respondentes possuem as seguintes estratégias:

(i) "*artor_org_A_selection_expert*" do agente envolvente "*artor_org_A*":
(i.i) o estilo do agente envolvente, presente na base de conhecimento corporativa:

```
/*************************************************************/
skill( style, artor_org_A, win_win).
/*************************************************************/
```

(i.ii) a estratégia e a táctica, presente na base de conhecimento individual do agente executor:

```
/*************************************************************/
frame (win_win, [
  strategy, [ decremental],
  tactics, [ troca],
```

```
    oper, [incrementar, decrementar, manter]]).
/*****************************************************/
```

(ii) "*artor_org_1_supply_expert*" do agente envolvente "*artor_org_1*":
(ii.i) o estilo do agente envolvente, presente na base de conhecimento corporativa:

```
/*****************************************************/
skill( style, artor_org_1, win_win).
/*****************************************************/
```

(ii.ii) a estratégia e a táctica, presente na base de conhecimento individual do agente executor:

```
/*****************************************************/
frame (win_win, [
  strategy, [ decremental_por_oferta],
  tactics, [ troca_oferta],
  oper, [incrementar, decrementar, manter]]).
/*****************************************************/
```

(iii) "*artor_org_2_supply_expert*" do agente envolvente "*artor_org_2*":
(iii.i) o estilo do agente envolvente, presente na base de conhecimento corporativa:

```
/*****************************************************/
skill( style, artor_org_2, win_win).
/*****************************************************/
```

(iii.ii) a estratégia e a táctica, presente na base de conhecimento individual do agente executor:

```
/*****************************************************/
frame (win_win, [
  strategy, [ decremental_por_oferta],
  tactics, [ troca_oferta],
  oper, [incrementar, decrementar, manter]]).
/*****************************************************/
```

A estratégia "*decremental*", implementada através da táctica de "*troca* ", baseada na "CL_P_I", utilizada pelo organizador no processo de selecção e de contratação de colaboradores, para a oferta inicial [fig. 72]:

(i) ***incrementar*** os valores que satisfazem os critérios, para seus valores de utilidade máxima (satisfaz mais).

"*artor_org_A_selection_expert*"					
critérios	utilidade (Pr_i)	valores que satisfazem	oferta inicial	valor relativo da instância	utilidade_ instância (Pr_i x valor relativo da instância)
cor	1	azul	azul	0	0
preço	3	20	10	+1	+3

período_de_ pagamento	3	30	120	+3	+9
quantidade	2	50	50	0	0

[fig. 72] Oferta inicial gerada pelo organizador

A utilidade da oferta inicial é calculada pelo somatório das utilidades das instâncias dos critérios:

$$\text{utilidade da oferta inicial} = \Sigma\ \text{utilidade_instância}_1, \ldots, \text{utilidade_instância}_n$$

a qual, para este caso é +12.

A estratégia "*decremental_por_oferta*", implementada pela táctica "*troca_oferta* ", baseada na "CL_P_i" utilizada pelos respondentes para a geração da contra-oferta [fig. 73]:

(i) ***manter*** o valor do critério, ofertado pelo organizador, o qual para o respondente possui a mais baixa utilidade (Pr_i). Caso exista mais de um, escolhe-se aleatoriamente um deles, ou mantém todos, e

(ii) ***incrementar ou manter*** os demais critérios em suas utilidades máximas (satisfaz mais).

"*artor_org_1_supply_expert*"				
critérios	utilidade (Pr_i)	oferta inicial	valor relativo da instância	utilidade_ instância (Pr_i x valor relativo da instância)
cor	2	preta	0	0
preço	1	10	-1	-1
período_de_ pagamento	3	vista	+1	+3
quantidade	3	80	+1	+3
"*artor_org_2_supply_expert*"				
critérios	utilidade (Pr_i)	oferta inicial	valor relativo da instância	utilidade_ instância (Pr_i x valor relativo da instância)
cor	2	azul	0	0
preço	3	30	+1	+3
período_de_ pagamento	2	120	-2	-4
quantidade	1	50	0	0

[fig. 73] Contra-oferta inicial gerada pelos respondentes

Para "***artor_org_1_supply_expert***" o critério preço é o de menor utilidade (Pr_i), motivo pelo qual, devido a sua estratégia, fixa-o e varia em suas utilidades máximas os critérios quantidade e período de pagamento.

Para "***artor_org_2_supply_expert***" os critérios quantidade e período de pagamento são os de menor utilidade (Pr_i), motivo pelo qual, devido a sua estratégia, fixa-os e varia em sua utilidade máxima o critério preço.

As utilidades das contra-ofertas iniciais são, respectivamente, para "*artor_org_1_supply_expert*" e "*artor_org_2_supply_expert*", +5 e -1.

3.4.3 A Metodologia de Negociação para a Resolução de Conflitos

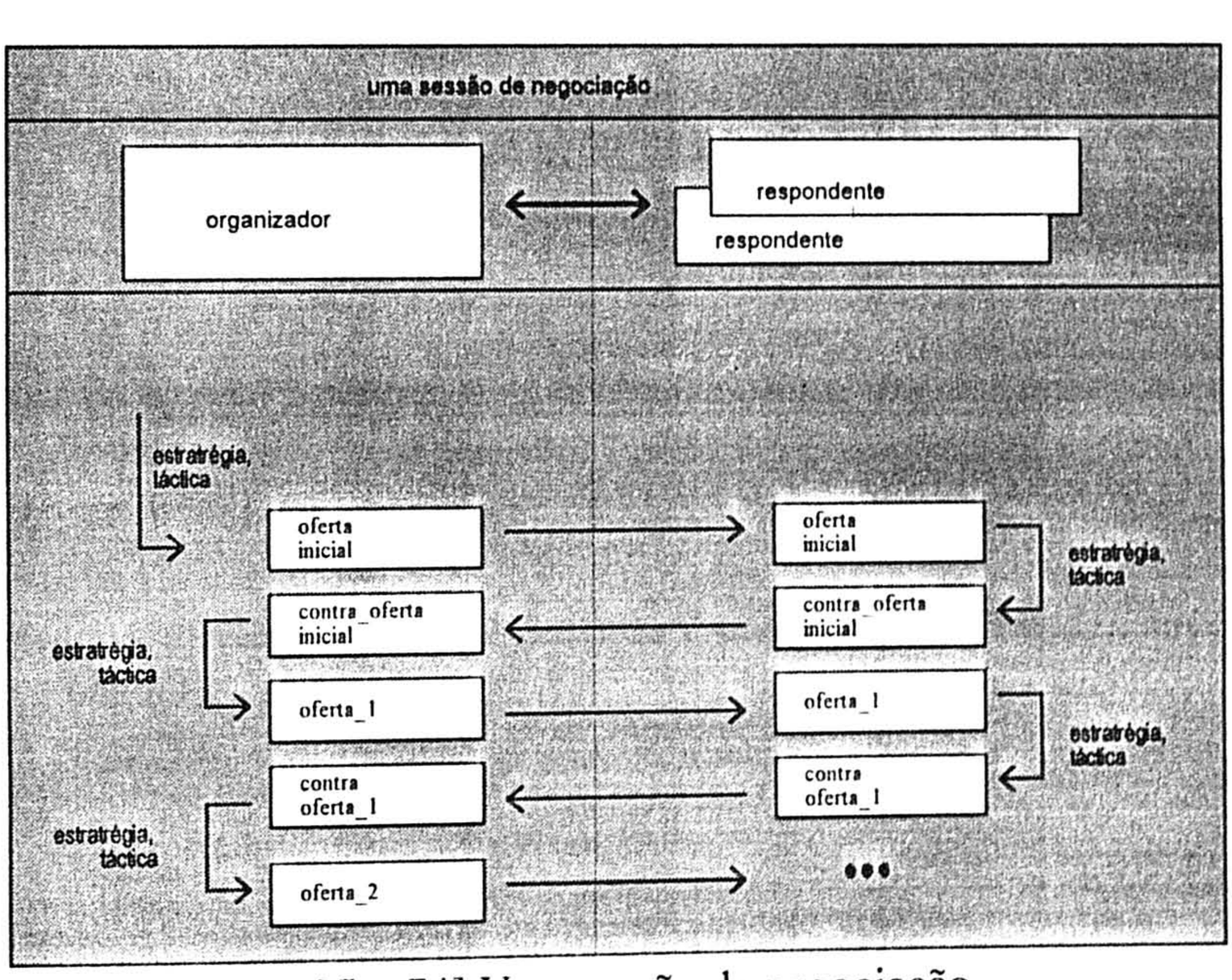
As diferenças encontradas pelo processo de avaliação, quando da comparação dos valores iniciais, ofertados e contra-ofertados, apresentam uma situação de conflito, a qual é tratada pela metodologia de negociação.

A metodologia de negociação utilizada, baseia-se em sucessivas gerações de ofertas e contra-ofertas [fig. 74], as quais constituem uma sessão de negociação.

Para que uma sessão de negociação encerre, há necessidade da existência de critérios de parada. Esses critérios são:

(i) pelo número de sessões previstas pelo organizador, quando do envio da oferta inicial, e/ou

(ii) por igualdade de valores de todos os critérios ofertados em relação aos contra-ofertados.



[fig. 74] Uma sessão de negociação

A exemplo da geração da oferta e das contra-ofertas iniciais, as demais ofertas e contra-ofertas são geradas através da aplicação da estratégia e da táctica escolhida pelo organizador e pelos respondentes.

Em referência ao cenário, a táctica de "*troca*" da estratégia "*decremental*", utilizada pelo organizador, passa a ser complementada com:

(i) para geração da oferta inicial:

(i.ii) ***incrementar*** os valores que satisfazem os critérios, para seus valores de utilidade máxima (satisfaz mais),

(ii) para as demais ofertas que não a inicial:

(ii.i) ***manter*** o valor de um dos critérios, o com mais alta utilidade ("Pr_i"). Caso exista mais de um, escolhe-se aleatoriamente um deles,

(ii.ii) seleccionar o próximo critério de mais alta utilidade e passar a ***decrementar***, de uma unidade, o valor relativo da instância, ***mantendo*** os demais fixos,

(i.ii.iii) quando todos os valores, do critério que está sendo decrementado, foram ofertados, e a sessão não foi encerrada, retornar ao passo (ii.ii), e

(i.ii.iv) quando não mais é possível manter o valor do critério escolhido no item (ii.i), **decrementar** de uma unidade, o valor relativo da instância desse critério, e retornar ao item (ii.ii).

Em referência ao cenário, a táctica de "*troca_oferta*" da estratégia "*decremental_por_oferta*", utilizada pelos respondentes, passa a ser complementada com:

(i) para geração da contra-oferta inicial:

(i.i) ***manter*** o valor do critério, ofertado pelo organizador, o qual para o respondente possui a mais baixa utilidade (Pr_i). Caso exista mais de um, escolhe-se aleatoriamente um deles ou mantém todos, e

(i.ii) ***incrementar ou manter*** os demais critérios em suas utilidades máximas (satisfaz mais),

(ii) para as demais contra-ofertas que não a inicial:

(ii.i) executar o item (i.i),

(ii.ii) ***decrementar***, a cada contra-oferta, de uma unidade, o valor relativo das instância dos critérios que tiveram suas utilidades posicionadas nos valores máximos.

As mensagens enviadas pelo organizador, contendo as ofertas, aos respondentes possuem o seguinte formato:

```
/*************************************************/
ask_offer( news_id( News_id),
           news_subject( News_subject),
           ask_content( Ask_content)
           ).
```

onde:

"*News_id*" é a identificação da notícia afixada,

"*News_subject*" é o título da notícia,

"*Askr_content*" contém a oferta:

<Message_number, Offer>

onde:

Message_number: é um cardinal, o qual contém o número de sequência da interacção,

Offer: contém a lista critérios com seus valores de oferta:

[Id,Vr, ...]

onde:

Id é a identificação do critério, e

Vr é o valor do critério

/**********/

As mensagens enviadas pelos respondentes, ao organizador, contendo as contra-ofertas, são:

(i) do mesmo formato utilizado para responder ao convite, e

(ii) o parâmetro "*Message_number*" contém o cardinal identificador da sequência da interacção.

As figuras [75a e 75b] a seguir, apresentam o resultado de todas as ofertas e contra-ofertas da sessão de negociação, para o problema enunciado no cenário, do ponto de vista do organizador "*artor_org_A_selection_expert*".

Nas figuras [75a e 75b] as:

(i) colunas "oferta ..." contém os valores gerados pelo organizador e enviados aos respondentes,

(ii) colunas "contra-oferta ..." contém os valores gerados pelos respondentes e recebidos pelo organizador, e

(iii) "*1*"e "*2*" correspondem, respectivamente, aos respondentes "*artor_org_1_supply_expert*" e "*artor_org_2_supply_expert*".

	oferta inicial		contra-oferta inicial		segunda oferta		segunda contra-oferta		terceira oferta		terceira contra-oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	azu	azu	pre	azu	pre	azu	pre	azu	pre	azu	pre	azu
preço	10	10	10	30	10	10	10	30	10	10	10	30
período_de_ pagamento	120	120	vis	120	90	90	30	90	60	60	60	60
quantidade	50	50	80	50	50	50	80	50	50	50	80	50

[fig. 75a] Resultados das ofertas e contra-ofertas da sessão de negociação - 1ª parte

	quarta oferta		quarta contra-oferta		quinta oferta		quinta contra-oferta		sexta oferta		sexta contra-oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	pre	pre	pre	azu	pre		pre		pre		pre	
preço	20	20	20	20	20		20		20		20	
período_de_ pagamento	120	120	vis	120	90		30		60		60	
quantidade	50	50	50	50	50		50		50		50	

[fig. 75b] Resultados das ofertas e contra-ofertas da sessão de negociação -2ª parte

O conhecimento das interacções são armazenados nas respectivas bases de conhecimento individual ("*Ikb*"), no formato das mensagens recebidas.

O organizador, através do convite enviado ao "*News_Stand*", fixou em seis (6) interacções para uma sessão de negociação, caso ocorresse discordância de pelo menos um dos valores dos critérios.

A aplicação da táctica de "*troca_oferta*" da estratégia "*decremental_por_oferta*", utilizada pelos respondentes, levou à estados distintos na geração das contra-ofertas, i.e.:

(i) o respondente "***artor_org_1_supply_expert***", antecipou o encerramento da sessão de negociação em relação ao "***artor_org_2_supply_expert***", por igualdade nos valores dos critérios, e

(ii) o respondente "***artor_org_2_supply_expert***", teve o encerramento da sessão de negociação, por dois critérios de parada, igualdade de valores e número de interacções previstas.

3.4.4 A Contratação

A contratação de um respondente, por um organizador, segue a seguinte prioridade:

(i) o respondente que apresentou a contra-oferta com a melhor utilidade, independente de ter sido a última interacção. Em caso de empate,

(ii) o respondente que antecipou o encerramento da sessão, por igualdade de valores. Em caso de empate,

(iii) o respondente cujo conhecimento que o organizador possui, em sua base de conhecimento, sobre processos de selecção anteriores, apontam como um respondente "win/win".

A mensagem enviada aos respondentes, de conclusão de processo de selecção e de contratação de colaboradores, possui o seguinte formato:

```
/*****************************************************/
  comfirming_process( news_id( News_id),
        news_subject( News_subject),
        process_acknowledgment( Process_acknowledgment)
        ).
```

onde:

Process_acknowledgment = {contracted , not_contracted}

```
/*****************************************************/
```

Em relação ao cenário, os resultados [fig. 76a e 76b] obtidos pelo organizador, no processo de selecção e de contratação de colaboradores, referente as utilidades apresentadas pelos respondentes foram:

	oferta inicial		contra-oferta inicial		segunda oferta		segunda contra-oferta		terceira oferta		terceira contra-oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	azu	azu	pre	azu	pre	azu	pre	azu	pre	azu	pre	azu
preço	10	10	10	30	10	10	10	30	10	10	10	30
período_de_ pagamento	120	120	vis	120	90	90	30	90	60	60	60	60
quantidade	50	50	80	50	50	50	80	50	50	50	80	50

utilidade das ofertas e contra-ofertas para o organizador	+12	+12	-2	+6	+9	+9	+1	+3	+6	+6	+6	0

[fig. 76a] Resultado obtido pelo organizador, referente ao processo de Selecção e de Contratação de Colaboradores - 1ª parte

	quarta oferta		quarta contra-oferta		quinta oferta		quinta contra-oferta		sexta oferta		sexta contra-oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	pre	pre	pre	azu	pre		pre		pre		pre	
preço	20	20	20	20	20		20		20		20	
período_de_pagamento	120	120	vis	120	90		30		60		60	
quantidade	50	50	50	50	50		50		50		50	
utilidade das oferta e contra-ofertas para o organizador	+9	+9	-3	+9	+6		0		+3		+3	

[fig. 76b] Resultado obtido pelo organizador, referente ao processo de Selecção e de Contratação de Colaboradores - 2ª parte

A obtenção dos valores da utilidade das ofertas e das contra-ofertas para o organizador é obtido pela fórmula:

$$\text{utilidade_instância} = (Pr_i \times \text{valor relativo da instância})$$

Utilizando as prioridades para a contratação, o respondente "*artor_org_2_supply_expert*", obteve:

(i) a maior pontuação de toda a sessão de negociação (+9), e

(ii) a maior pontuação ocorreu na quinta contra-oferta, encerrando por igualdade de valores, a sessão.

As mensagens enviadas pelo organizador aos respondentes foram:

(i) para "*artor_org_1_supply_expert*":

```
comfirming_process( news_id(seleccao_contratacao),
        news_subject(desenvolvimento_de_produto_pl),
        process_acknowledgment( not_contracted)
        ).
```

(ii) para "*artor_org_2_supply_expert*",

```
comfirming_process( news_id(seleccao_contratacao),
        news_subject(desenvolvimento_de_produto_pl),
        process_acknowledgment( contracted)
        ).
```

3.4.5 Aprendendo com o Processo de Negociação

O processo de negociação, possibilita, para os agentes que dele participaram, aprender a partir das interacções.

Uma sessão de negociação é geradora de um conjunto de exemplos, resultado das ofertas e das contra-ofertas entre os organizadores e os respondentes.

A aprendizagem em "*ARTOR*" é composta das seguintes fases:

(i) preparação do conjunto de exemplos,

(ii) raciocínio indutivo, e

(iii) raciocínio dedutivo.

A preparação do conjunto de exemplos, consiste da análise dos valores gerados pelas ofertas e contra-ofertas atribuindo aos mesmos, por critério, se mantém, incrementam ou decrementam a utilidade para o agente.

Quando a utilidade do valor do critério contra-ofertado:

(i) se mantiver em relação ao valor ofertado, então **mantém a utilidade**, representado por (man_util),

(ii) se elevar em relação ao valor ofertado, então **incrementa a utilidade**, representado por (incr_util), e

(iii) diminuir em relação ao valor ofertado, então **decrementa a utilidade**, representado por (decr_util).

Quando a utilidade de duas ofertas consecutivas, tomando como referencia para a inicial, os valores desejados dos critérios:

(i) se mantiver em relação ao valor ofertado anterior, então **mantém a utilidade**, representado por (man_util),

(ii) se elevar em relação ao valor ofertado anterior, então **incrementa a utilidade**, representado por (incr_util), e

(iii) diminuir em relação ao valor ofertado anterior, então **decrementa a utilidade**, representado por (decr_util).

Utilizando as interacções entre organizador e respondentes, do cenário apresentado[fig. 77a e 77b], teremos para cada:

(i) uma das contra-ofertas em relação as ofertas, e

(ii) para cada oferta em relação a oferta anterior,

os seguintes estados (os estados de mantém, incrementa e decrementa estão representados na figura pelos sinais de =, ↑e ↓, respectivamente):

	oferta inicial		contra-oferta inicial		segunda oferta		segunda contra-oferta		terceira oferta		terceira contra-oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	azu =	azu =	pre =	azu =	pre =	azu =	pre =	azu =	pre =	azu =	pre =	azu =
preço	10 ↑	10 ↑	10 =	30 ↓	10 =	10 =	10 =	30 ↓	10 =	10 =	10 =	30 ↓
período_de_ pagamento	120 ↑	120 ↑	vis ↓	120 =	90 ↓	90 ↓	30 ↓	90 =	60 ↓	60 ↓	60 =	60 =
quantidade	50 =	50 =	80 ↓	50 =	50 =	50 =	80 ↓	50 =	50 =	50 =	80 ↓	50 =

[fig. 77a] Estados das utilidades das ofertas e das contra-ofertas - 1ª parte

	quarta oferta	quarta contra-	quinta oferta	quinta contra-	sexta oferta	sexta contra-

			oferta				oferta				oferta	
critérios	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"	para "*1*"	para "*2*"	de "*1*"	de "*2*"
cor	pre =	pre =	pre =	azu =	pre =		pre =		pre =		pre =	
preço	20 ↓	20 ↓	20 =	20 =	20 =		20 =		20 =		20 =	
período_de_ pagamento	120 ↑	120 ↑	vis ↓	120 =	90 ↓		30 ↓		60 ↓		60 =	
quantidade	50 =	50 =	50 =	50 =	50 =		50 =		50 =		50 =	

[fig. 77b] Estados das utilidades das ofertas e das contra-ofertas -2ª parte

Transformando a atribuição dos estados de cada oferta e contra-oferta:

(i) em listas compostas de atributos e de seus valores, considerando um dos atributos, a classe do exemplo,
(ii) considerando para classe dos exemplos, os critérios e seus estados quando contra-ofertado, e
(iii) criando tantos conjuntos de exemplos quantos forem os critérios de uma lista de critérios (*CL*), que variaram durante uma sessão de negociação,

o organizador obterá os seguintes conjuntos produzidos pelas interacções, com:

(i) "*artor_org_1_supply_expert*":

os critérios que variaram durante a sessão de negociação, devido as contra-ofertas do respondente foram: quantidade e prazo. Os demais critérios se mantiveram.

```
----- EXEMPLOS ----------------------> classe = critério quantidade
[preco(incr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(decr_util_Qtde)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Qtde)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Qtde)]
[preco(decr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(man_util_Qtde)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(man_util_Qtde)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(man_util_Qtde)]

------ EXEMPLOS ----------------------> classe = critério prazo
[preco(incr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(decr_util_Prazo)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Prazo)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(man_util_Prazo)]
[preco(decr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(decr_util_Prazo)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Prazo)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(man_util_Prazo)]
```

(ii) "*artor_org_2_supply_expert*":

o critério que variou durante a sessão de negociação, devido as contra-ofertas do respondente foi o preço. Os demais critérios se mantiveram.

```
------ EXEMPLOS ----------------------> classe = critério preço
[preco(incr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(decr_util_Preco)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Preco)]
[preco(man_util),cor(man_util),quantidade(man_util),prazo(decr_util),classe(decr_util_Preco)]
[preco(decr_util),cor(man_util),quantidade(man_util),prazo(incr_util),classe(man_util_Preco)]
```

O organizador, aplicando o algoritmo de aprendizagem simbólica, de construção de árvores de decisão (Anexo F), para os exemplos acima apresentados, é obtido:

(i) relativo as contra-ofertas do "*artor_org_1_supply_expert*":

(i.i) para os exemplos com a classe relativa a quantidade:

```
------ TREE ------------------------->
  preco
    incr_util-->
      decr_util_Qtde
    man_util-->
  cor
    man_util-->
      quantidade
        man_util-->
          prazo
            decr_util-->
              man_util_Qtde
decr_util-->
  man_util_Qtde
```

(i.ii) para os exemplos com a classe relativa a prazo:

```
------ TREE ------------------------->
  preco
    incr_util-->
      decr_util_Prazo
    man_util-->
      cor
        man_util-->
          quantidade
            man_util-->
              prazo
                decr_util-->
                  man_util_Prazo
```

```
    decr_util-->
      decr_util_Prazo
```

(ii) relativo as contra-ofertas do "*artor_org_2_supply_expert*":

(ii.i) para os exemplos com a classe relativa a preço:

```
------ TREE ------------------------->
  preco
    incr_util-->
      decr_util_Preco
    man_util-->
      decr_util_Preco
    decr_util-->
      man_util_Preco
```

A aplicação da indução como aprendizagem simbólica, conduz o organizador a concluir que:

(i) o respondente "*artor_org_1_supply_expert*", concentra sua estratégia e táctica de negociação nos critérios quantidade e prazo. O critério preço é o critério relevante para haver a variação,

(ii) o respondente "*artor_org_1_supply_expert*", se utiliza de 50% dos critérios para negociar, e 50% para concordar,

(iii) o respondente "*artor_org_2_supply_expert*", concentra sua estratégia e táctica de negociação no critério preço, sendo o próprio preço o atributo mais relevante para haver variação, e

(iv) o respondente "*artor_org_2_supply_expert*", se utiliza de apenas 25% dos critérios para negociar, e 75% para concordar.

Em relação ao organizador, essas conclusões são obtidas com o seguinte comportamento:

. para cada agente respondente,
. para cada um dos critérios das contra-ofertas que apresentam variações durante a sessão de negociação,
. gerar uma árvore de decisão, tendo como classe, o critério da contra-oferta que apresentou a variação,
. para cada árvore de decisão gerada, obter o nó raiz. Este nó é um critério relevante, causador das variações nas contra-ofertas,

. *os demais critérios que não variaram, são critérios através dos quais os repondentes relaxam suas restrições, concordando com os mesmos.*

As conclusões obtidas são utilizadas para a criação de conhecimento sobre os respondentes e armazenadas na base de conhecimento individual ("*ikb*") do organizador.

```
/*****************************************************/
respondente (artor_org_1_supply_expert).
concorda_criterios(artor_org_1_supply_expert, 50).
discorda_criterios(artor_org_1_supply_expert, 50).
criterio_oferta_relevante_variacao(
                artor_org_1_supply_expert, [preco]).
criterio_contra_oferta_relevante_variacao(
                artor_org_1_supply_expert,[quantidade,
                prazo]).
contratado(artor_org_1_supply_expert, nao).

respondente (artor_org_2_supply_expert).
concorda_criterios(artor_org_2_supply_expert, 75).
discorda_criterios(artor_org_2_supply_expert, 25).
criterio_oferta_relevante_variacao(
                artor_org_2_supply_expert, [preco]).
criterio_contra_oferta_relevante_variacao(
                artor_org_2_supply_expert,[preco]).
contratado(artor_org_2_supply_expert, sim).
/*****************************************************/
```

3.5 Conclusões

As relações interorganizações para o provimento de bens/serviços, em uma sociedade de organizações, podem ser vistas como relações sociais de cooperação baseadas em compromissos, as quais contribuem no alcance dos objectivos organizacionais das partes envolvidas.

Numa sociedade de organizações, quando os componentes desempenham papéis de consumidores, significa que eles estão interessados em serem providos de bens/serviços. Quando os componentes desempenham papéis de distribuidores, significa que eles estão interessados em serem os provedores dos bens/serviços.

Para que estas relações se concretizem, uma das possibilidades, a tratada por este capítulo, é quando a iniciativa para o provimento de um bem/serviço é tomada pela organização que está desempenhando o papel de consumidora, sendo as demais possíveis fornecedoras.

As organizações para assegurarem o provimento de um bem/serviço, necessitam decidir sobre critérios que permitam avaliar o perfil de um potencial provedor. Para conhecer os potenciais provedores, o processo de selecção e de contratação de colaboradores foi adoptado.

O processo de selecção e de contratação de colaboradores está baseado em uma lista de critérios que expressa as dimensões que os potenciais fornecedores serão avaliados. Devido a diversidade de comportamentos e interesses, os potenciais fornecedores podem não concordar com os valores dos critérios, gerando uma situação de conflito. Para que o impasse gerado pela situação de conflito possa ser resolvido, foi utilizada a negociação como metodologia de resolução.

A negociação é, fundamentalmente, um processo de tomada de decisão. A negociação tem sido abordada em várias disciplinas [96, 97], tais como a teoria dos jogos, a economia , as relações internacionais e laborais e a psicologia. Ela tem recebido a atenção de investigadores nos domínios da Inteligência Artificial Distribuída [97, 98, 99, 100], bem como no desenvolvimento de modelos teóricos no qual a aprendizagem desempenha um importante papel.

No modelo proposto, por este capítulo, a negociação se baseia no estilo do agente, em estratégias e em tácticas, para decidir e gerar novos valores para os critérios de avaliação. Estes novos valores são respostas aos componentes de uma negociação, quando se apresenta a situação de conflito. A geração de novos valores para os critérios de avaliação, formam um conjunto de exemplos, os quais, representam o comportamento de um fenómeno. Estes exemplos são utilizados no processo de aprendizagem simbólica, contribuindo para que novos conhecimentos, sobre o comportamento dos componentes de um processo de selecção e de contratação de colaboradores, possam ser aprendidos.

Capítulo 4

CONCLUSÕES

A área de Inteligência Artificial e as suas componentes Inteligência Artificial Distribuída - Sistemas Multiagente e Aprendizagem Simbólica Automática, apresentam paradigmas que suportam a modelação de problemas inerentemente distribuídos do mundo real, tal como é o caso das organizações numa sociedade de organizações.

As organizações são constituídas por entidades distribuídas, tanto em termos de funcionalidades quanto em termos de espaço. Podemos olhar para cada uma das suas unidades funcionais, como uma unidade cognitiva capaz de executar actividades que concorram para o alcance do objectivo global. Esta cognição sustenta a possibilidade de tomar decisões, executar tarefas dentro da especialidade e aprender.

Pensar num modelo para as organizações como se fossem sistemas monolíticos seria, além de negar a distribuição, característica que lhe é própria, atribuir-lhes um tamanho tal que comprometeria a fiabilidade, a compreensão e a flexibilidade, além de dotá-los de uma grande complexidade.

Modelá-las como se fossem sistemas distribuídos de controlo centralizado seria conduzi-las a um paradigma que traria problemas devido a elevada concentração de decisões a serem tomadas por um único componente. Para além disso, iríamos descaracterizá-las quanto à autonomia, (mesmo que no mundo real esta autonomia seja parcial), que cada unidade funcional possui. A modelação deste problema através de metodologias de Sistemas Multiagente e de Aprendizagem Simbólica Automática, prova ser adequada na medida em que:

(i) se aproxima do modo como na realidade este problema é formulado,
(ii) diminui a complexidade dos componentes, uma vez que cada unidade funcional constitui um módulo,
(iii) aumenta a flexibilidade de alterações, inclusões, eliminações e movimentações das unidades funcionais,
(iv) permite uma maior eficiência, usufruindo do partilhamento do tempo através do paralelismo,
(v) permite um tratamento a falhas quando existe mais de uma unidade funcional com capacidade de executar a tarefa,
(vi) permite o balanceamento da carga computacional através da distribuição dos componentes de "*software*" [54],

(vii) baseada em Aprendizagem Simbólica Automática, cada unidade funcional passa a usufruir das experiências proporcionadas pela execução das actividades de sua competência, através da aprendizagem empírica, e contribui para incremento do conhecimento corporativo.

Uma característica fundamental que difere o interior ("*inner*") de uma organização do seu exterior ("*outer*"), encontra-se na questão relativa à coordenação. No interior de uma organização os agentes que dela participam possuem diferentes graus de autonomia na execução de suas actividades, na negociação ou na argumentação. As solicitações não podem ser recusadas sem uma legítima razão, pois os objectivos globais da organização não podem ser comprometidos. Por outro lado, no exterior de uma organização, cada agente da sociedade (organização ou indivíduo) é autónomo nas suas decisões (salvaguardando as regras, tais como as advindas da autoridade constituída, por exemplo o Estado), agindo em seu próprio benefício, motivado pelos seus interesses.

Os ambientes exteriores onde os agentes de uma sociedade multiagente interagem podem ser classificados com descentralizados e centralizados. O critério fundamental para esta classificação está centrado nos mecanismos de coordenação, caracterizados pela distribuição da decisão e por um maior ou menor esforço nas comunicações.

O sistema "*ARTOR*", proposto por esta dissertação, permite modelar uma sociedade de organizações, baseado em metodologias de Inteligência Artificial Distribuída, mais especificamente as de Sistemas Multiagente. O objectivo último do sistema "*ARTOR*" é orientar a implementação computacional do modelo das organizações, permitindo uma automatização de relações inter e intra-organizações.

Uma sociedade de organizações, para o sistema "*ARTOR*" possui os seguintes componentes:

(i) as organizações,

(ii) um quiosque, local público de afixação de notícias, e

(iii) uma interface homem/artefacto.

As organizações são modeladas por agentes denominados de agentes envolventes. Em um agente envolvente as competências para exercer as actividades requeridas pelos sub-domínios que constituem uma organização, são distribuídas em dois tipos de agentes:

(i) os dotados de competência para gerir a organização, chamados de agentes gestores, e

(ii) os dotados de competência para executar as especialidades requeridas pela organização, chamados de agentes executores.

Esta distribuição de competências permite explorar a execução das capacidades dos agentes de forma modular, estruturada, paralela e com diferentes graus de autonomia. Concentra a execução da especialidade nos agentes executores, modelando-os sem capacidades de planeamento e de coordenação de suas competências perante a organização. Por sua vez, os agentes gestores, dedicam-se à satisfação dos objectivos organizacionais, gerindo de forma integrada as competências das especialidades a eles vinculadas, num certo período.

O outro componente de um agente envolvente, a base de conhecimento corporativa, representa o conhecimento que não pertence, em particular, a nenhum dos agentes que compõem a organização. O conhecimento corporativo é adquirido, aprendido e mantido pelos agentes internos ao agente envolvente, em função das suas

intra e interrelações sociais. O conhecimento corporativo é orientador do comportamento dos agentes gestores e executores no desempenho de suas competências, tendo como finalidade que as actividades a serem executadas estejam de acordo com os objectivos, intenções e perfil da organização.

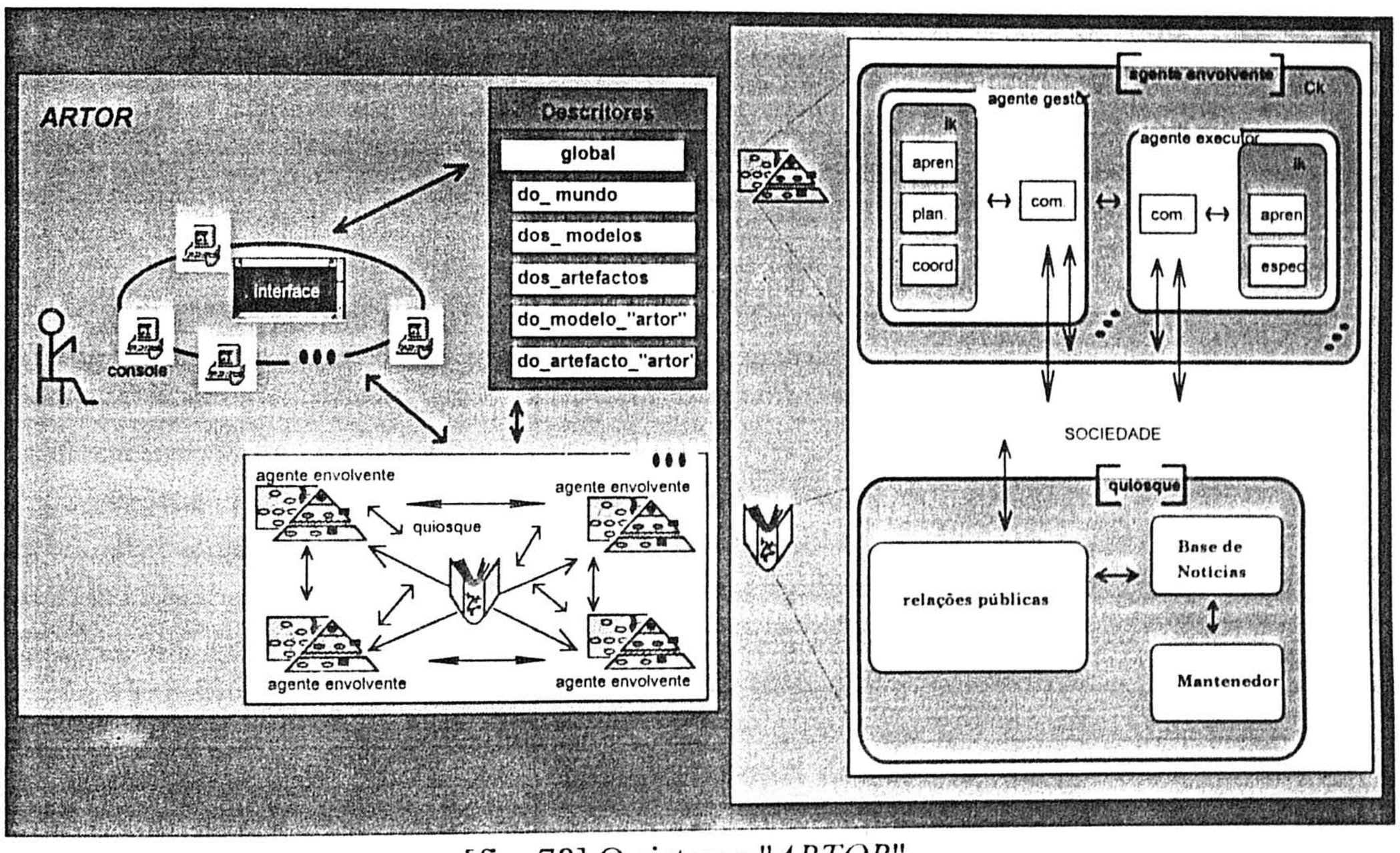
O quiosque, local público de afixação de notícias, tem por função disponibilizar conhecimentos sobre factos ou eventos sociais para os agentes envolventes que estão interessados em oportunidades de negócios. Sua implementação é baseada em metodologias de "quadro-negro", possibilitando a afixação, de notícias, a gestão de suas veiculações e a disponibilização das mesmas quando solicitadas.

A interface homem/artefacto permite ao utilizador visualizar as trocas de mensagens entre os componentes intra e interagente envolventes. Permite inquirir os agentes envolventes quer quanto as suas actividades quer sobre o seu conhecimento.

Para suportar a modelação de uma sociedade de organizações, baseada em recursos computacionais, que permita de uma maneira flexível e incremental implementar os componentes e as relações desta sociedade, o sistema "*ARTOR* [fig. 78] utiliza:

(i) descritores parametrizados pelo utilizador, e

(ii) mecanismos genéricos de geração de componentes da sociedade.



[fig. 78] O sistema "*ARTOR*"

Os descritores, descrevem e estruturam o conhecimento sobre os modelos e artefactos relacionados com uma sociedade de organizações. São utilizados para a geração e para o lançamento dos componentes, bem como para suportarem as explicações no domínio.

Os mecanismos genéricos de geração de componentes, possibilitam que, de forma dinâmica, as capacidades e as metodologias que implementam as competências dos componentes sejam activadas.

Diferentes aproximações de modelação, no domínio das organizações [64, 66, 67, 68], recorrem a sistemas baseados em recursos computacionais, utilizando

metodologias de Inteligência Artificial Distribuída, e/ou Sistemas Multiagente. O modelo apresentado por esta dissertação, buscou a generalização na construção de artefactos que suportem as diferentes estruturas e actividades organizacionais. Baseia-se na divisão das competências para a gestão e para execução de uma especialidade nos vários sub-domínios de actuação de uma organização, apresentando os seguintes benefícios:

(i) eficiência pelo partilhamento do tempo, permitido pelo paralelismo,
(ii) expansibilidade, possível pela agregação de novos agentes, seja na comunidade, seja intra agente envolvente,
(iii) modularidade, através do decréscimo de complexidade dos componentes intra agente envolvente,
(iv) compreensão de uma organização, possível pelo particionamento da competência global de uma organização e pela modularidade, e
(v) reutilização, pois os componentes podem ser utilizados na construção de outros agentes envolventes.

Quanto a distribuição, os componentes (agentes) do sistema "*ARTOR*" podem ser classificados:

(i) do ponto de vista intra-agente envolvente, os componentes são distribuídos e de controlo variando, entre a total descentralização e a total centralização, dependendo da estrutura organizacional adoptada, e
(ii) do ponto de vista da sociedade, os componentes são também vistos como distribuídos e de controlo descentralizado.

As relações interorganizações para o provimento de bens/serviços, em uma sociedade de organizações, podem ser vistas como relações sociais de cooperação baseadas em compromissos, as quais contribuem no alcance dos objectivos organizacionais das partes envolvidas.

Numa sociedade de organizações, quando os componentes desempenham papéis de consumidores, significa que eles estão interessados em serem providos de bens/serviços. Quando os componentes desempenham papéis de distribuidores, significa que eles estão interessados em serem os provedores dos bens/serviços.

Para que estas relações se concretizem, uma das possibilidades, a tratada pelo Capítulo III, é quando a iniciativa para o provimento de um bem/serviço é tomada pela organização que está desempenhando o papel de consumidora, sendo as demais possíveis fornecedoras.

As organizações para assegurarem o provimento de um bem/serviço, necessitam decidir sobre critérios que permitam avaliar o perfil de um potencial provedor. Para conhecer os potenciais provedores, o processo de selecção e de contratação de colaboradores foi adoptado.

O processo de selecção e de contratação de colaboradores está baseado em uma lista de critérios que expressa as dimensões que os potenciais fornecedores serão avaliados. Devido a diversidade de comportamentos e interesses, os potenciais fornecedores podem não concordar com os valores dos critérios, gerando uma situação de conflito. Para que o impasse gerado pela situação de conflito possa ser resolvido, foi utilizada a negociação como metodologia de resolução.

A negociação é, fundamentalmente, um processo de tomada de decisão. A negociação tem sido abordada em várias disciplinas [96, 97], tais como a teoria dos jogos, a economia , as relações internacionais e laborais e a psicologia. Ela tem

recebido a atenção de investigadores nos domínios da Inteligência Artificial Distribuída [97, 98, 99, 100], bem como no desenvolvimento de modelos teóricos no qual a aprendizagem desempenha um importante papel.

No modelo proposto em "*ARTOR*", a negociação se baseia no estilo do agente, em estratégias e em tácticas, para decidir e gerar novos valores para os critérios de avaliação. Estes novos valores são respostas aos componentes de uma negociação, quando se apresenta a situação de conflito. A geração de novos valores para os critérios de avaliação, formam um conjunto de exemplos, os quais, representam o comportamento de um fenómeno. Estes exemplos são utilizados no processo de aprendizagem simbólica, contribuindo para que novos conhecimentos, sobre o comportamento dos componentes de um processo de selecção e de contratação de colaboradores, possam ser aprendidos.

4.1 Limitações

O estudo das organizações e a modelação e implementação por computador, constitui um trabalho infindável. Fundamentalmente, o limite dessa dissertação possui seu contorno situado nas relações intra e interorganizações.

Do ponto de vista das relações intra-organização foram investigadas as estruturas e elaborado um modelo que possa implementá-las através de Sistemas Multiagente e Aprendizagem Simbólica Automática. Foi investigado e implementado a relação social de selecção e de contratação de colaboradores, cuja parte desta relação também se encontra interna à organização. Nesta relação social, a capacidade de aprendizagem encontra-se muito aquém do potencial que oferece.

Outras relações intra-organização merecem ser investigadas, dadas as particularidades que cada uma pode apresentar, por exemplo a cadeia de comando interna e a comunicação informal.

Do ponto de vista interorganizações a relação social de selecção e de contratação de colaboradores proporcionou a utilização de diversos recursos computacionais (metodologias, "*software*" e "*hardware*"), os quais potencialmente, se apresentam úteis para a implementação de outras relações no domínio das organizações.

4.2 Trabalhos Futuros

Em relação as actuais funcionalidades do sistema "*ARTOR*", poderíamos ter como melhoramento:

(i) na relação social de selecção e de contratação de colaboradores:

(i.i) critérios de selecção condicionais, i.e. restrições que limitem o espaço de possibilidades de negociação utilizando experiências passadas, e

(i.ii) utilização do conhecimento aprendido, durante o processo de negociação, com o objectivo de, em tempo real, alterar as estratégias e tácticas utilizadas para gerar as ofertas e as contra-ofertas,

(ii) nas estruturas das organizações:

(ii.i) tornar as estruturas das organizações dinâmicas, adequadas aos novos objectivos.

Em relação a novas funcionalidades do sistema "*ARTOR*", poderíamos:

(i) investigar a relação social de lançamento no mercado de novos produtos, gerando os meios de divulgação do produto, adequados ao consumidor alvo.

Referências

[1] Simon, H.A., The Sciences of The Artificial, Massachusetts, The M.I.T. Press, 1968. p.1-22.

[2] Gopnik, M., Cognitive Sciences. Em: Encyclopedia of Physical Science and Technology. Academic Press, Inc., 1987. p.123-139.

[3] Chiavenato, I., Introdução à Teoria Geral da Administração. São Paulo, Editora McGraw-Hill Ltda, 1993. p.473-479.

[4] Malone, T.W., Modeling Coordination in Organizations and Markets. Em: Reading in Distributed Artificial Intelligence. Editado por Alan H.Bond and Les Gasser, Morgan Kaufmann Publishers, inc, San Mateo, CA,1988.

[5] Fox, M. S., An Organizational View of Distributed Systems. Em IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-11, No 1, January 1981. p.70-80.

[6] Russel, S., Norvig P., Artificial Intelligence - A Moderm Approach. Prentice Hall, Inc. 1995.pg 4-5.

[7] Brooks A. R., Intelligence Without Reason. Em: 12th International Joint Conference on Artificial Intelligence, August 1991.

[8] Barr, A., Feigenbaum, E. A., The Handbook of Artificial Intelligence, vol I. Pitman Books Limited, 1981.

[9] Bond, A. H.; Gasser, L., An Analysis of Problems an Research in DAI. Em: Readings in Distributed Artificial Intelligence. Editado por Alan H.Bond and Les Gasser, Morgan Kaufmann Publishers, inc, San Mateo, CA, 1988.

[10] Mouta, F., Ambiente de Desenvolvimento de Sistemas Multiagente para Controlo e Supervisão de Processos em Aplicações Distribuídas. Dissertação para obtenção do grau de doutor, Faculdade de Engenharia, Universidade do Porto, 1996.

[11] Moulin, M.; Chaib-Draa, B., An Overview of Distributed Artificial Intelligence. Em Foundations of Distributed Artificial Intelligence. Editado por G. M. P. O'Hara and N. R. Jennings, Foundations of Distributed Artificial Intelligence, John Wiley & Sons, inc., 1996.

[12] Durfee, E.,H.; Rosenschein, J, S., Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. Em Proceedings of The International Workshop on Distributed Artificial Intelligence, July 1994.

[13] Franklin, S.; Graesser, A., Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. Em European Conference on Artificial Intelligence - ECAI' 96, Budapest, Hungary, 1996.

[14] Maes, P., Artificial Life Meets Entertainment: Life like Autonomous Agents. Em Communications of the ACM, 38, 11, 1995.

[15] Smith, D. C.; Cypher, A.; Spohrer,J., KidSim: Programming Agents without a Programming Language. Em Communications of the ACM, 37, 7, 1994.
[16] Hayes-Toth, B., An Architecture for Adaptive Intelligent Systems. Em Arificial Intelligence: Special Issue on Agents and Interactivity, 72, 1995.
[17] Wooldridge, M.; Jennings, N. R., Agent Theories, Architectures, and Languages: A Survey. Em Intelligent Agents. Editado por Wooldridge. M. and Jennings, N. R., Berlim, Spring-Verlag, 1995.
[18] Rao, A. S.; Georgeff, M. P., BDI Agents: From Theory to Practice. Technical Note 56, Australian Artificial Intelligence Institute, April, 1995.
[19] Wooldridge, M.; Jennings, N. R., Intelligent Agents: Theory and Practice. Em Knowledge Engineering Review, October 1994.
[20] Davidsson, P.; Autonomous Agents and the Concept of Concepts. Doctoral dissertation, Department of Computer Science, Lund University, Sweden, 1996.
[21] Maes, P.; Social Interface Agents: Acquiring competence by learning from users and other agents. Em Etzioni, O., editor, Software Agents - Papers from the 1994 Spring Symposium, Technical Report SS-94-03,p. 71-80, AAAI Press, 1994.
[22] Engelmore, R. S.; Morgan, A. J.; Nii, H.P., Introduction. Em Engelmore, R.; Morgan, T. editor, Blackboard Systems, p. 1-22. Addison-Wesley Publishing Company, 1988.
[23] Chang, E.; Participant Systems. Em Huhns, M., editor, Distributed Artificial Intelligence, p- 311-340. Pitman Publishing: London and Morgam Kaufmann: São Mateo, CA, 1987.
[24] Demazeau, Y.; From Interactions to Collective Behaviour in Agent-Based Systems. Em European Conference on Congnitice Sciences, St-Malo, France, 1995.
[25] Craig, I; Blackboard Systems. Ablex Publishing Corporation, New Jersey-USA, 1995.
[26] Maes, P.; The agent network architecture (ANA). SIGART Bulletin, 2(4), p.115-120,1991.
[27] Etzione, O. ; Lesh, N. ; Segal, R. ; Building Softbots for UNIX. Em Etzione, O., editor. Software Agents, p.9-16. AAAI Press, 1994.
[28] Goleman, D. ; Inteligência Emocional, Círculo de Leitores, 1996.
[29] Rosemblatt, F., The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, vol. 65, p. 386-407, 1958.
[30] Selfridge, O. G., Pandemonium: A Paradigm for Learning, Proceedings of the Symposium on Mechanization of Thought Processes, Blake, D. and Uttley, A. editores, HMSO, London, p. 511-529, 1959.
[31] Widrow, B., Generalization and Information Storage in Networks of Adelaine 'Neurons', Spartan Books, Washington, D. C., p. 435-461, 1962.
[32] Samuel, A. L., Some Studies in Machine Learning Using the Game of Checkers, IBM journal of Research and Development, No. 3, p. 211-229, 1959.
[33] Samuel, A. L., Some Studies in Machine Learning Using the Game of Checkers, Computers and Thought, Feigenbaum, E. A. and Feldman, J. editores, McGraw-Hill, NY, p. 71-105, 1963.
[34] Buchanan, B. G. and Mitchell, T. M., Model-Directed Learning of Production Rules, Pattern Directed Inference Systems, Waterman, D. A. and Hayes-Roth, F. editores, Academic Press, NY, 1978.
[35] Buchanan, B. G., Feigenbaum, E. A. and Lederberg, J., A Heuristic Programming Study of Theory Formation in Sciences, Proceedings of the Second International Joint Conference on ARtificial Intelligence, International Joint Conferences on Artificial Intelligence, London, p. 40-48, 1971.

[36] Winston, P. H., Learning Structural Descriptions from Examples, The Psychology of Computer Vision, Winston, P. H. editor, McGraw Hill, NY, ch. 5, 1975.

[37] Simon, H. A., Why Should Machines Learn?, em Machine Learning: An Artificial Intelligence Approach, R. S. Michalski, J. G., and T. M. Mitchell, editores, Tioga, Palo Alto, California, 1983.

[38] Minsky, M., The Society of Mind, MIT Press, Cambridge, 1985.

[39] Michalski, R. S., Understanding the Nature of Learning: Issues and Research Directions. Em R.S. Michalski, J. G. Carbonell, and T. M. Mitchell ,editores, Machine Learning: An Artificial Intelligence Approach, Volume II. Los altos, California, Morgan Kaufmann Publishers, Inc, 1986.

[40] Carbonell, J. G., Introduction: Paradigms for Machine Learning. Em J. G. Carbonell editor, Machine Learning: Paradigms and Methods, MIT Press, 1990.

[41] Michalsky, R. S., Carbonell, J. G., and Mitchell, T.M. editores, Machine Learning: An Artificial Intelligence Approach, Morgan Kaufmann, 1983.

[42] Michalsky, R. S., Carbonell, J. G., and Mitchell, T.M. editores, Machine Learning: An Artificial Intelligence Approach, Volume II, Morgan Kaufmann, 1986.

[43] Michalsky, R. S., Kodratoff, Y. editores, Machine Learning: An Artificial Intelligence Approach, Volume III, Morgan Kaufmann, 1990.

[44] Biermann, A. W., Guiho, G., and Kodratoff, Y., Automatic Program Construction Techniques, MacMillan Publishing Company, NY, 1984.

[45] Winston, P.H., Learning New Principles from Precedents and Exercises, em Artificial Intelligence 19, p. 321-350, 1982.

[46] Burstein, M. H., Concept Formation by Incremental Analogical Reasoning and Debugging, em Machine Learning: An Artificial Intelligence Approach, Volume II, Michalski, R.S., Carbonell, J.G.,Mitchell, T.M., editores, Morgan Kaufmann, 1986.

[47] Carbonell, J.G., Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition, em Machine Learning: An Artificial Intelligence Approach, Volume II, Michalski, R.S., Carbonell, J.G.,Mitchell, T.M., editores, Morgan Kaufmann, 1986.

[48] Michalsky, R. S., Integrated Multistrategy Learning: Toward a Unified Theory of Learning, Reports of Machine Learning and Inference Laboratory, MLI90-1, Center for Artificial Intelligence, George Mason University, 1990.

[49] Shaw, M. J., Mechanisms for Cooperative Problem Solving and Multi-agent Learning in Distributed Artificial Intelligence Systems, em Proceedings of the 10th International Workshop on Distributed Artificial Intelligence, Bandera, Texas, 1990.

[50] Shaw, M. J., Whinston, A. B., Learning and Adaptation in Distributed Artificial Intelligence Systems, em Distributed Artificial Intelligence, Vol. II editores Gasser, L., Huhns, M.N., Morgan Kaufmann Publishers, 1989.

[51] Russel, S., Norvig P., Artificial Intelligence - A Moderm Approach. Prentice Hall, Inc. 1995.pg 526.

[52] Sian, S. S., Extending Learning to Multiple Agents: Issues and a Model for Multi-Agent Machine Learning (MA-ML), European Working Session on Learning-EWSL, 1990.

[53] Durkin, J., Expert Systems Catalog of Applications, Intelligent Computer Systems Inc. Akron, Ohio, 1993.

[54] Oliveira, E., Mouta, F., Parallel Execution through Expertise Distribution, Proceedings of the Cost 229 WG4 Workshop on Massively Parallel Computing, Universidade da Madeira, Porto, 1993.

[55] Hammer, M., Champy, J., Reengineering the Corporation: A Manifesto for Business Revolution, Nicholas Brealey Publishing Limited, London, 1994.
[56] Mintzberg, H., Estrutura e Dinâmica das Organizações, Publicações Dom Quixote, Portugal, 1995.
[57] Fox, M. S., An Organizational View of Distributed Systems, em IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-11, No.1, 1981.
[58] Simon, H. A., Decision Making and Organizational Design, em Organization Theory, Edited by D. S. Pugh, Penguin Books, 1960.
[59] Ghosal, A., Cognitive Revolution in Management, em Kybernetes-The International Journal os Systems & Cybernetics, edited by Rudall, B. H., Vol. 22, No. 6, MCB University Press Limited,1993.
[60] Malone, T. W.; Crowston, K., The Interdisciplinary Study of Coordination, em ACM Computing Surveys, Vol.26, No.1, 1994.
[61] Singh, B., Rein, G. L., Role Interaction Nets (RIN): A process description formalism. Tech. Rep. CT-083-92. Microelectronics and Computer Technology Corp., Autin, Tex,1992.
[62] Malone, T. W.; Crowston, K., Toward an Interdisciplinary Theory of Coordination. Tech. Rep. no 120. Massachusetts Institute of Technology, Center for Coordination Science, Cambridge, Mass, 1991.
[63] Oliveira, E., Mouta, Fernando, Rocha, A. P., Negotiation and Conflict Resolution Within a Community of Cooperative Agents, em Proceedings of the First International Symposium on Autonomous Decentralized Systems-ISADS93, Kawasaki, Japão,1993.
[64] Skarmeas, N., A Framework for Multi-Agent Modeling and Office Automation, em The Second International Conference on The Practical Application of Prolog, editado por Leon Sterling, Londres, 1994.
[65] Cosmadopoulos, Y., Chu, D. A., IC prolog II reference manual. Technical report, Logic Programming Section, Imperial College, London, 1993.
[66] Barbuceanu, M., Fox, M. S., The Information Agent: An Infrastructure for Collaboration in the Integrated Enterprise, Second International Working Conference on Cooperating Knowledge based Systems, University of Keele, editor Deen, S.M., 1994.
[67] Norman, T. J., Jennings, N. R., Faratin, P., Mamdani, E. H., Designing and implementing a multi-agent architecture for business process management, 12th European Conference on Artificial Intelligence - ECAI'96, Hungary, 1996.
[68] Jennings, N. R., Faratin, P., Johnson, M. J., O'Brien, P., Wiegand, M. E., Using Intelligent Agents to Manage Business Processes, The First Internacional Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology - PAAM' 96, London, UK,1996.
[69] So, Y., Durfee E. H., An Organizational Self-Design Model for Organizational Change, Workshop Notes the AAAI-93 Workshop on Artificial Intelligence and Theories of Groups and Organizations, 1993.
[70] Corkill, D. D., Lesser V. R., The use of meta-level control for coordination in a distributed problem solving network, em Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983.
[71] Gasser, L., Ishida, T., A dynamic organizational architecture for adaptive problem solving, em Proceedings of the National Conference on Artificial Intelligence, 1991.

[72] Durfee, E. H., Montgomery, T. A., Coordination as distributed search in a hierarchical behavior space, em IEEE Transactions on Systems, Man, and Cybernetics, 1991.
[73] Oliveira, E., Mouta, F., Rocha, A. P., Cooperation in a Multi-Agent Community, Proceedings The Thirteenth International Avignon Conference on Artificial Intelligence Tools, Techniques, Methods and Applications, France, 1993.
[74] Oliveira, E., Ramos, C., A Cooperative Multi-Agent System for an Assembly Robotic Cell, em Advanced Robotics, Vol. 10, No.1, The International Journal of the Robotics Society of Japan, Fukuda, ,T. editor, Nagoya University, Japan,1996.
[75] Wellman, M. P., A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems, em Jornal of Artificial Intelligence Research 1, 1993.
[76] Davis, R., Smith, R. G., Negotiation as a Metaphor for Distributed Problem Solving, em: Reading in Distributed Artificial Intelligence. Editado por Alan H.Bond and Les Gasser, Morgan Kaufmann Publishers, inc, San Mateo, CA,1988.
[77] Kuhn, N., Müller, H. J., Müller, J. P., Task Decomposition in Dynamic Agent Societies, em Proceedings of the International Symposium on Autonomous Decentralized Systems (ISADS-93), Tokyo, Japan, IEEE Computer Society Press, 1993.
[78] Fischer, K., Kuhn, N., Müller, H. J., Müller, J. P., Pischel, M., Sophisticated and Distributed: The Transportation Domain, em Pre-Proceedings of The 5th European Workshop on Modelling Autonomous Agents in a Multi_agent World-MAAMAW'93, Neuchatel, 1993.
[79] Dictionary of Science and Technology, Academic Press, Inc, 1992.
[80] Mowshowitz, A., Social Dimensions of Office Automation, Advances in Computers,25, 1986.
[81] Davidow, W. H., Malone, M. S., The Virtual Corporation: Structuring and Revitalizing The Corporation for 21st Century, New York: Harper Business, 1992.
[82] Quinn, J. B., Hilmer, F. G., Strategic outsourcing, The McKinsey Quarterly,1, 1995.
[83] Scheer, A.-W., Business Process Engineering - Reference Models for Industrial Enterprises, Spring-Verlag, Berlin, 2nd edition, 1994.
[84] Müller, P. J., Fischer, K., Heiming, I., Scheer, A-W., Intelligent Agents in Virtual Enterprises, The First Internacional Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology - PAAM' 96, London, UK,1996.
[85] Müller, P. J., An Architecture for Dynamically Interacting Agents, PhD thesis, Universität des Saarlandes, Saarbrücken, 1996.
[86] Rabelo, J. R., Camarinha-Matos, L. M., Control and Dynamic Scheduling in Virtual Organization of Production Resources, em International Conference on Evaluation of Production Management Methods, Porto Alegre, Brasil, 1994.
[87] Rabelo, J. R., Camarinha-Matos, L. M., A Distributed Architecture based on Negotiation for Dynamic Scheduling in FMS, 4th Portuguese Conference on Computer Aided Production / Project / Planning, Lisbon, Portugal, 1993.
[88] Kim, H. M., Fox, M. S., Formal Models of Quality and ISO9000 Compliance: An Information Systems Approach, em American Quality Control Conference, American Society for Quality Control, Las Vegas, NV, 1994.
[89] Fonseca, J. M., Oliveira, E., Steiger-Garção, A., A DAI Based Resource Management System, The First Internacional Conference and Exhibition on The

Practical Application of Intelligent Agents and Multi-Agent Technology - PAAM' 96, London, UK, 1996.
[90] Gasser, L., Hulthage, I., Leverich, B., Organizations as Complex, Dynamic Design Problems, Progress in Artificial Intelligence 6th Portuguese Conference on AI-EPIA'93, edited by Filgueiras, M., Damas, L. Springer-Verlag, Porto, Portugal, 1993.
[91] Hardwick, M., Spooner, D. L., Rando, T., Morris, K. C., Sharing Manufacturing Information in Virtual Enterprises, Communications of The ACM, Vol. 39, No 2, 1996.
[92] Damas, L., Costa, V. S., Azevedo, R., Reis, R., YAP Reference Manual, Centro de Informática da Universidade do Porto, 1990.
[93] Shmeil, M. A. H., Oliveira, E., Detecting the Opportunities of Learning from the Interactions in a Society of Organizations, 12th Brazilian Symposium on Artificial Intelligence - SBIA'95, Campinas, Brazil, 1995.
[94] Oliveira, E., Shmeil, M.A.H., ARTOR - An Approach towards ARTificial Organizations, International Symposium on the Management of Industrial and Corporate Knowledge - ISMICK'95, Compiègne, France, 1995.
[95] Osborne, M. J.,Rubinstein, A., A Course in Game Theory. The MIT Press, 1994.
[96] Rosenschein, J., Zlotkin, G., Rules of Encounter. MIT Press, Cambridge, 1994.
[97] Oliveira, E., Fonseca, J.M., Steiger-Garção, A., Negotiation between Economically Motivated Agents, em Balanced Automation Systems II. Proceedings of the 2nd IEEE/ECLA/IFIP- International Conference on Architectures and Design Methods for Balanced Automation Systems, Porto, Portugal. Edited by Camarinha-Matos, L.M., Afsarmanesh, H., Chapman & Hall, 1996.
[98] Mor, Y., Goldman, C. V., Rosenschein, J. S., Learn Your Opponent's Strategy (in Polynomial Time)!. Em Proceedings of IJCAI'95, Workshop on Adaptation and Learning in Multiagent Systems, 1995.
[99] Sandholm, T., Lesser, V., Issues in Automated Negotiation and Eletronic Commerce: Extending the Contract Net Framework. Em Proceedings First International Conference on Multiagent Systems, San Francisco, 1995.
[100] Chu-Caroll, J., Carberry, S., Conflict Detection and Resolution in Collaborative Planning. Em Wooldridge, M., Müller, J. P., Tambe, M., editors, Intelligent Agents, Workshop on Agent Theories, Architectures and Languages- ATAL'95.
[101] Sandholm, T. W., Lesser, V. R., Coalition formation among bounded rational agents. Em Proceedings of 14th International Joint Conference on Artificial Intelligence, 1995.
[102] Zeng, D., Sycara, K., How can an Agent Learn to Negotiate. Em 12th European Conference on Artificial Intelligence - ECAI'96, Workshop on Agent Theories, Architectures and Languages, Budapest, Hungary, 1996.
[103] Pearl, J., Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann Publishers, Inc., 1988.
[104] Rosenschein, J., Zlotkin, G., Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. MIT Press, Cambridge, 1994.
[105] Müller, J. P., A Cooperation Model for Autonomous Agents. Em Müller, J., Wooldridge, M., Jennings, N., editors, Workshop on Agent Theories, Architectures and Languages, Budapest, Hungary, 1996.
[106] Wooldridge, M., Bussmann, S., Klosterberg, M., Production Sequencing as Negotiation. Em The First International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'96, 1996.

[107] Oliveira, E., Shmeil, M.A.H., Applying Multiagent Methodologies to Organization Modeling, V Congreso Iberoamericano de Inteligencia Artificial - Iberamia, Cholula, Puebla, México, Noviembre de 1996.
[108] Cockayne, W.T., Zyda, M., Mobile Agents, Manning Publications Co, 1998.
[109] Huhns, M.N., Munindar, P.S.,editores, Readings in Agents, Morgan Kaufmann, Inc, 1998.
[110] Shoham, Y., Agent-oriented Programming, Artificial Intelligence, 60(1):51-92, 1993.
[111] Haugeneder, H., Steiner, D., McCabe, F.G., IMAGINE: A Framework for Building Multi-Agent Systems, Proceedings of The 2nd International Conference on Cooperative Knowledge Based Systems, Keele University, Deam, S. M., editor, 1994.
[112] White, J. E., Telescript Technology: The Foundation for the Eletronic Marketplace, white paper, General Magic, Inc, Latham Street, Mountain View, CA, 1994.
[113] Bigus, J.P., Bigus, J., Constructing Intelligent Agents with JAVA: A Programmer's Guide to Smarter Applications, John Wiley & Sons, Inc., 1998.
[114] Han, J., Chiang, J.Y., Chee, S., Chen, J., Chen, Q.,Cheng, S., Gong, W., Kamber, M., Kopersky, K., Liu, G., Lu, Y.,Stefanovic, N., Winstone, L., Xia, B. B.,Zaine, O. R., Zhang, S., Zhu, H., Dbminer: A System for Data Mining in Relational Databases and Data Warehouses, em Proceedings of The International Conference on Data Mining and Knowledge Discovery - KDD'96, Oregon, 1996.
[115] Kamber, M., Winstone, L., Gong, W., Cheng, S.,Han, J., Generalization and Decision Tree Induction: Efficient Classification in Data Mining, em Proceedings of Internatonal Workshop on Research Issues on Data Engineering - RIDE'97, Birmingham, England, 1997.
[116] Vinoski, S., CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, em IEEE Communications Magazine, Vol 14, nº 2, 1997.
[117] Chalupsty, H., Finin, T., Fritzson, R., Mckay, D., Shapiro, S., Wiederhold, G., An Overview of "*KQML*": A Knowledge Query and Manipulation Language (Draft), Stanford University, Computer Sacience Departament, 1992.
[118] Finin T., Fritzson, R., Mckay, D., McEntire, R., "*KQML*" - A Language and Protocol for Knowledge and Information Exchange, Proceedings of the 13th International Distributed Artificial Intelligence Workshop, Seattle WA, 1994.
[119] Oliveira E., Qiegeng, L., Camacho, R., Controlling Cooperative Experts in a Real-Time System, em Proceedings of the Third International Conference on Software Engineering for Real Time Systems, Cirencester, UK, 1991.
[120] Wittig, T., ARCHON: Architecture for Cooperative Multi-Agent Systems, Ellis Horwood, 1992.
[121] Emerson, E.A. Temporal and Modal Logic, em Leewen, J. V., editor, Handbook of Theoretical Computer Science, Volume B, Fomal Models and Semantics, pages 995-1072, Elsevier Science Publishers and MIT Press, Amsterdam and Cambridge, MA, 1990.
[122] Kinny, D., Georgeff M., Commitment and Effectiveness of Situated Agents, em Proceedings of The Twelfth International Joint Conference on Artificial Intelligence - IJCAI'91, pages 82-88, Sydney, Austrália, 1991.
[123] Rao, A., S., Georgeff, M. P., Modeling Rational Agents within a BDI-Architecture, Technical Notes 14, Autralian Artificial Intelligence Institute, 1991.

[124] Oliveira, E., Mouta, F., UPShell, a Shell for Building Multi-Agent Systems, em Proceedings of the Second Compulog Net Area Meeting on Knowledge Bases, Atenas, Grécia, 1993.
[125] Finin, T. Fritzson, R., Mckay, D., McEntire, R., "*KQML*" as an Agent Communication Language, Third International Conference on Information and Konwledge Management - CIKM'94, AACM Press, 1994.
[126] Genesereth, M. R., Knowledge Intechange Format. em Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning. 599-600, 1991.
[127] Elmasri, R., Navathe, S., Fundamentals of Database Systens, second edition. Benjamin/Cummings, Redwood City, CA, 1994.
[128] Austin, J. L., How to Do Things with Words. Clarendon Press, Oxford, 1962
[129] Goodwin, R., Formalizing Properties of Agents, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1993.
[130] Fikes, R. E., Nilsson, N., STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, Artificial Intelligence, 5(2):189-208, 1971.
[131] Rolston, D.W., Principles of Artificial Intelligence and Expert Systems Development, McGraw-Hill International editions, 1988.
[132] Sacerdoti, E., Planning in a Hierarchy of Abstraction Spaces, Artificial Intelligence, 5:115-135, 1974.
[133] Sacerdoti, E., The Non-linear Nature of Plans, em Proceedings of the Fourth International Joint Conference on Artificial Intelligence - IJCAI'75, Stanford, CA, 1975.
[134] Ambros-Ingerson, J., Steel, S., Integrating Planning, Execution and Monitoring, em Proceedings of The Seventh National Conference on Artificial Intelligence - AAAI'88, St. Paul, Mn, 1988.
[135] Wood, T., Planning and Decision Making in Dynamic Domains, Ellis Horwood, Chichester, England, 1993.
[136] Cohen, P.R., Geenberg, M.L.,Hart, D.M., Howe, A.E., Trial by Fire: Understanding The Design Requirements for Agents in Complex Environments, AI Magazine, 10(3):32-48, 1989.
[137] Pollack, M.E., Ringuette, M., Introducing The Tileworld: Experimentally Evaluating Agent Architectures. em Proceedings of The Eighth National Conference on Artificial Intelligence - AAAI'90, Boston, MA, 1990.
[138] Vere, S., Bickmore, T., A Basic Agent, Computational Intelligence, 6:41-60, 1990.
[139] Brooks, R.A., Intelligence without Representation, Artificial Intelligence, 47:139-159, 1991.
[140] Georgeff, M.P., Lansky, A.L., Reactive Reasoning and Planning, em Proceedings of The Sixth National Conference on Artificial Intelligence - AAAI'87, Seatttle, WA, 1987.
[141] Ferguson, I. A., Towards an Architecture for Adaptive, Rational, Mobile Agents, em Werner, E., Demazeau, Y., editors, Decentralized AI 3 - Proceedings of The Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds - MAAMAW'91, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1992.
[142] Burmeister, B., Sundermeyer K., Cooperative Problem Solving Guided by Intentions and Perception, em Werner, E., Demazeau, Y., editors, Decentralized AI 3 - Proceedings of The Third European Workshop on Modelling Autonomous Agents

and Multi-Agent Worlds - MAAMAW'91, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1992.
[143] Haddadi, A., A Hybrid Architecture for Multi-agent Systems. In Deen, S.M., editor, em Proceedings of The 1993 Workshop on Cooperating Knowledge Based Systems - CKBS'93, DAKE Centre, University of Keele, UK, 1993.
[144] Smith, R.G., A Framework for Distributed Problem Solving, UMI Research Press, 1980.
[145] Quinlan, J. R., Induction of Decision Trees, Machine Learning 1, 1986.
[146] Quinlan, J. R. Discovering Rules by Induction from Large Collections of Examples. Expert Systems in the Micro Electronic Age, D. Michie editor, Edinburgh University Press, Edimburgo, 1979.
[147] Quinlan, J. R., Compton, P.J., Lazarus, L., Inductive Knowledge Acquisition: A Case Study, em Proceedings of the 2nd Autralian Conference on Applications of E.S., Sydney, 1986.
[148] Mingers, J., An Empirical Comparison of Selection Measures for Decision Tree Induction. Machine Learning 3, 1989.
[149] Cestenik, B.; Kononenko, I.;Bratko, I. Assistant 86: A Knowledge Elicitation Tool for Sophisticated Users, Progress in Machine Learning, Bratko, I., Lavrac, N., editors, Sigma Press, 1987.
[150] Nilsson, N.J., Principles of Artificial Intelligence, Palo Alto, CA, Tioga, 1980.
[151] Dejong, G., Mooney R., Explanation-Based Learning: An Alternative View, Machine Learning 1, 1986.
[152] Flann, N.S., Dietterich, T.G., A Study of Explanation-Based Methods for Inductive Learning; Machine Learning 4, 1989.
[153] Terabe, M., Washio, T, Katai, O., Sawaragi, T., A Study of Organizational Learning in Multi_Agent Systems, em Distributed Artificial Intelligence Meets Machine Learning, Weiβ, G., editor, European Conference on Artificial Conference - ECAI'96, Springer , 1997.
[154] Shmeil, M.A.H., Oliveira, E., The Establishment of Partnerships to Create Virtual Organizations: A Multiagent Approach, International Conference on Integrated and Sustainable Industrial Production - ISIP'97, Lisboa - Portugal, 1997.
[155] Quinlan, J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, Califórnia, 1993.

ANEXOS

Anexo A

Sistemas "*BDI*" ("*Beliefs, Desires and Intentions*")

O estudo de agentes computacionais, capazes de apresentar um comportamento racional, tem recebido grande atenção da comunidade científica.

Os projectos de sistemas computacionais que requerem alto nível de controlo de tarefas em ambientes dinâmicos e complexos, estão tornando-se um importante diferencial (incremental) comercial. Estes sistemas incluem a gestão do controlo de tráfego aéreo, redes de telecomunicações, processos de negócios, veículos espaciais e serviços médicos. As aplicações de técnicas convencionais (sistemas de informação), para desenvolver estes sistemas, são complexas e caras em suas elaborações, verificações e manutenções.

Uma das arquitecturas [fig. A1], para um sistema distribuído que atenda o exposto, é a modelada através de agentes racionais dotados de atitudes mentais de **crenças, desejos e intenções** ("*Beliefs, Desires and Intentions*") [123], que representam, respectivamente, a informação, a motivação e os estados deliberativos do agente.

As **crenças** de um agente correspondem a informação que o agente possui sobre o mundo, a qual pode ser completa ou incompleta. Os **desejos** de um agente, intuitivamente, correspondem as tarefas atribuídas para ele. As **intenções** de um agente representam desejos comprometidos para a execução. A intenção é que um agente não seja, em geral, capaz de alcançar todos os seus desejos, mesmo que esses desejos sejam consistentes. O agente, entretanto, fixa um subconjunto de desejos disponíveis e compromete seus recursos para o alcance deles. Os desejos escolhidos são as intenções. Um agente, tipicamente, continuará tentar alcançar uma intenção até que, ou sua crença sobre a intenção esteja satisfeita, ou ele crê que a intenção não mais será alcançada.

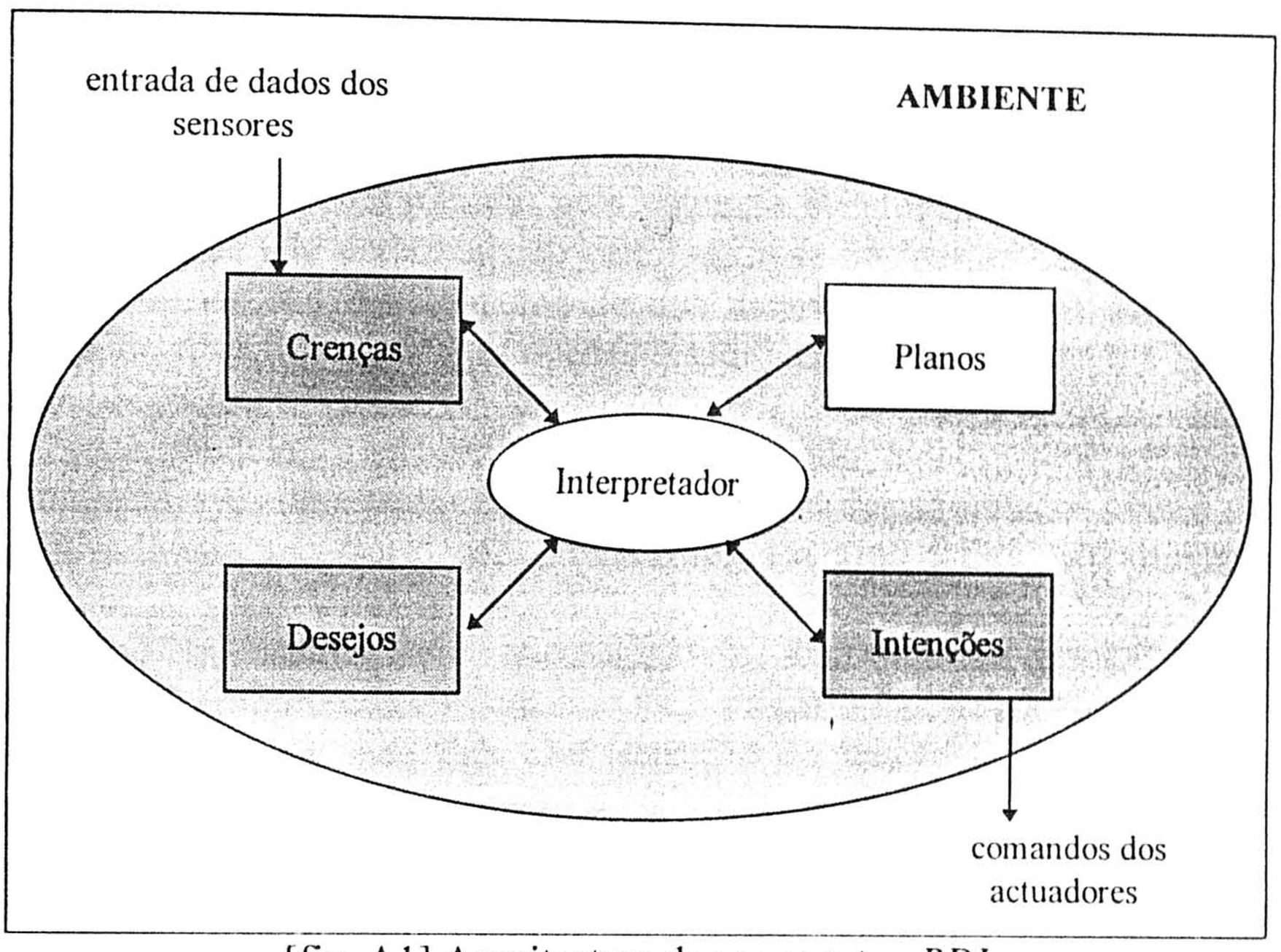
Uma estrutura complementar de dados em um sistema "*BDI*" é uma biblioteca de planos ("*plan library*"). Uma biblioteca de planos é um conjunto de planos que especifica o curso de acção que pode ser seguido pelo agente para o alcance das intenções. Uma biblioteca de planos de um agente representa seu conhecimento em forma de procedimentos, i.e. o "saber fazer".

Um plano contém duas partes:

(i) um corpo ou programa, o qual define um curso de acção, e

(ii) um descritor que especifica as pré e pós condições de utilização de um plano.

A arquitectura de um sistema "*BDI*", apresenta mais um componente, o interpretador ("*interpreter*"). O interpretador é o responsável pela actualização das crenças a partir das observações efectuadas no mundo. Ele gera novos desejos (tarefas) com base nas novas crenças, e selecciona a partir do conjunto corrente dos desejos activos, algum subconjunto para actuar como intenção. Finalmente, o interpretador selecciona uma acção para ser executada pelo agente, com base nas intenções e no conhecimento disposto em forma de procedimentos.



[fig. A1] Arquitectura de um agente "*BDI*"

Uma vasta classe de aplicações em tempo real, exibem um número de importantes características, que estabelecem a necessidade de modelos baseados em crenças, desejos e intenções:

(a) em qualquer instante no tempo, existe potencialmente muitos diferentes caminhos nos quais o ambiente pode desdobrar-se (o ambiente é não determinístico). Como exemplo, em um cenário de um aeroporto, quando uma aeronave está a aterrar, a direcção do vento pode mudar, outras aeronaves podem estar presentes, a condição da pista pode ser alterada, etc,

(b) em qualquer instante no tempo, existe potencialmente, diferentes acções ou procedimentos que o sistema pode executar,

(c) em qualquer instante no tempo, existe potencialmente diferentes objectivos que o sistema é requerido a executar,

(d) as acções e procedimentos que suportam os vários objectivos a serem alcançados, são dependentes do estado do ambiente e são independentes do estado interno do sistema. Para o exemplo do aterrar

de uma aeronave, a direcção do vento é uma variável do ambiente, independente do estado do sistema,
(e) o ambiente pode apenas ser percebido localmente, i.e. a percepção de uma das variáveis pode não ser suficiente para determinar o estado de todo o ambiente, e
(f) o ambiente é dinâmico, gerando a necessidade que o sistema responda (tempo de processamento) adequadamente frente as mudanças.

Para modelar o comportamento de um sistema que suporte as características mencionadas, encontramos em [121] para as características (a) e (b), uma estrutura em árvore, onde cada ramo desta, representa um caminho alternativo de execução. Cada nó da estrutura representa um certo estado do ambiente e cada transição uma primitiva de acção a ser executada pelo sistema, uma primitiva de evento ocorrendo no ambiente, ou ambas.

As múltiplas acções tomadas pelo sistema e eventos que ocorrem no ambiente, diferentes tipos de não determinismo manifestam-se de "*per se*" em diferentes tipos de nós. Estes nós são denominados de nós de escolha (decisão) e nós de troca, representando respectivamente, as opções disponíveis para o sistema e a incerteza do ambiente.

Neste modelo, os objectivos do sistema são identificados com caminhos particulares, através da estrutura em árvore, cada um deles rotulados com o objectivo a ser realizado e se necessário, com o beneficio ou retorno obtido pelo atravessamento do caminho.

Como o sistema tem que agir, ele necessita seleccionar acções ou procedimentos apropriados a serem executados a partir das várias opções disponíveis. Para um projecto para este fim, uma selecção de função deve possibilitar que o sistema alcance, efectivamente, seus objectivos a partir dos recursos computacionais disponíveis e das características do ambiente no qual o sistema está situado.

Para a selecção de uma função, sob o domínio das características mencionadas, existem ao menos dois tipos de entrada de dados requeridas:

(i) dada a característica (d), é essencial que o sistema tenha informações do estado do ambiente. Este tipo de componente do sistema é denominado de crença ("*belief*"), o qual pode ser implementado como uma variável, uma base de dados, um conjunto de expressões lógicas, ou alguma outra estrutura de dados. As crenças podem ser vistas como componentes informativos de um estado do sistema, e
(ii) é necessário que o sistema também tenha informação sobre os objectivos a serem realizados, ou mais genericamente, que as prioridades ou os retornos estejam associados com os vários objectivos correntes (características (c) e (d)). Este tipo de componente é denominado de desejo ("*desire*"), o qual pode representar o estado de motivação do sistema.

A aproximação mais relevante para a modelação de uma função de selecção é a teoria da decisão, contudo, esta teoria não leva em consideração a característica (f), nomeadamente, que o ambiente possa ser alterado de forma significativa durante a execução da função de selecção ou durante a execução do curso de acção, determinada pela função de selecção.

A presença da primeira situação pode ser reduzida através da utilização da mais rápida, computacional, função de selecção (talvez a não óptima). A segunda possibilidade, a teoria clássica da decisão e a ciência clássica da computação providenciam diferentes respostas. A teoria da decisão indica o reaplicar da função de selecção em um ambiente alterado, e a ciência clássica da computação, um programa uma vez iniciado, aguarda o seu término sem qualquer avaliação de sua utilidade.

Nenhuma das aproximações mencionadas satisfaz a característica (f), pois tanto no reaplicar da função de selecção quanto no aguardar do término de um ciclo de processamento, o dispêndio de tempo e de recursos, incrementa o risco de ocorrência de trocas significativas no ambiente, levando o sistema a não alcançar o objectivo ou o retorno esperado. Considerando que a escolha de troca de acções a cada passo de execução é muito dispendiosa e a escolha da acção tem a possibilidade de não ser válida, é assumido que trocas potencialmente significativas podem ser instantaneamente determinadas (em um nível de granulosidade definida pelas acções primitivas e eventos do domínio). Este procedimento conduz a criação de limites para a frequência das reconsiderações e então alcança um equilíbrio apropriado entre muitas e poucas reconsiderações [122]. Para que este tratamento seja possível, é necessário incluir um componente no estado do sistema para representar o actual curso e acção escolhida, i.e. o resultado da mais recente chamada para a função de selecção. A este componente do sistema denomina-se de intenções ("*intentions*"), o qual em essência captura o componente deliberativo ("*deliberative*") do sistema.

Anexo B

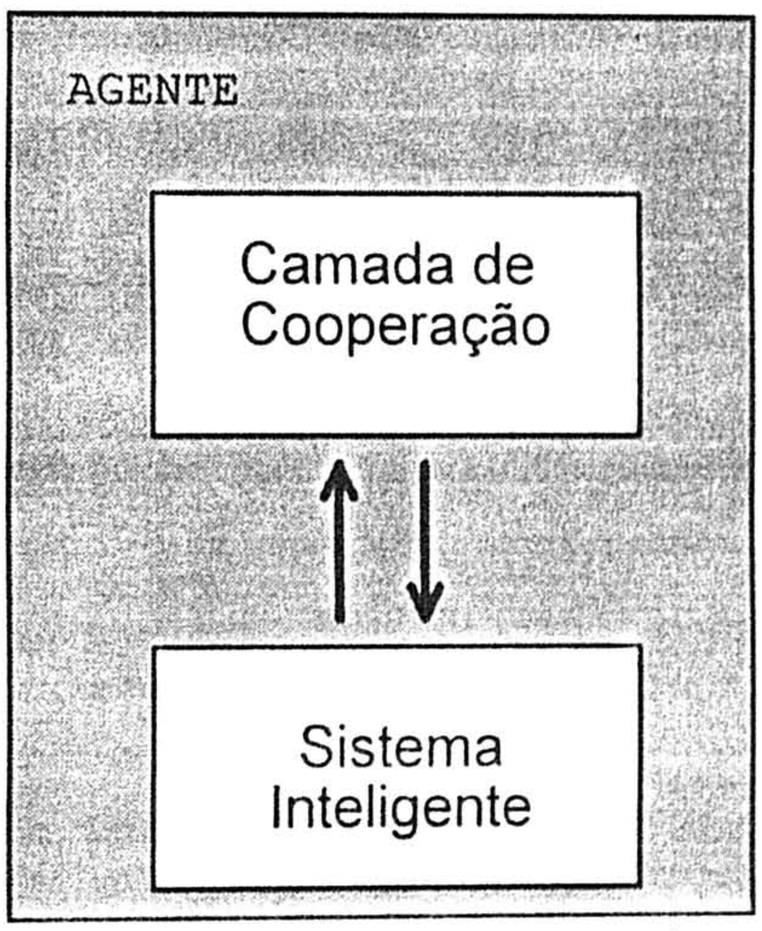
Linguagens de Comunicação Intra e Interagentes
"UPShell" ("University of Porto Shell")

O sistema "*UPShell*" ("***University of Porto Shell***"), desenvolvido por Eugénio Oliveira e Fernando Mouta [73, 124], para auxiliar o desenvolvimento de sistemas multiagentes próprios para aplicações industrias, tem como principal objectivo, gerar ou transformar sistemas inteligentes distribuídos já existentes, numa comunidade de agentes cooperativos. Os sistemas ditos inteligentes são sistemas periciais ou outros sistemas baseados em conhecimento, com a capacidade de resolução de problemas num domínio específico.

Os agentes gerados pelo "*UPShell*" são constituídos, fundamentalmente, por dois módulos [fig. A2]:

(i) o sistema inteligente, e

(ii) a camada de cooperação, a qual controla a actividade do sistema inteligente e a cooperação com os demais componentes de uma comunidade de agentes.



[fig. A2] - Arquitectura de um agente gerado pelo "*UPShell*"

A comunicação Intra e Interagentes

Enquanto o sistema inteligente resolve os problemas do domínio, a camada de cooperação é a responsável pelas seguintes actividades:

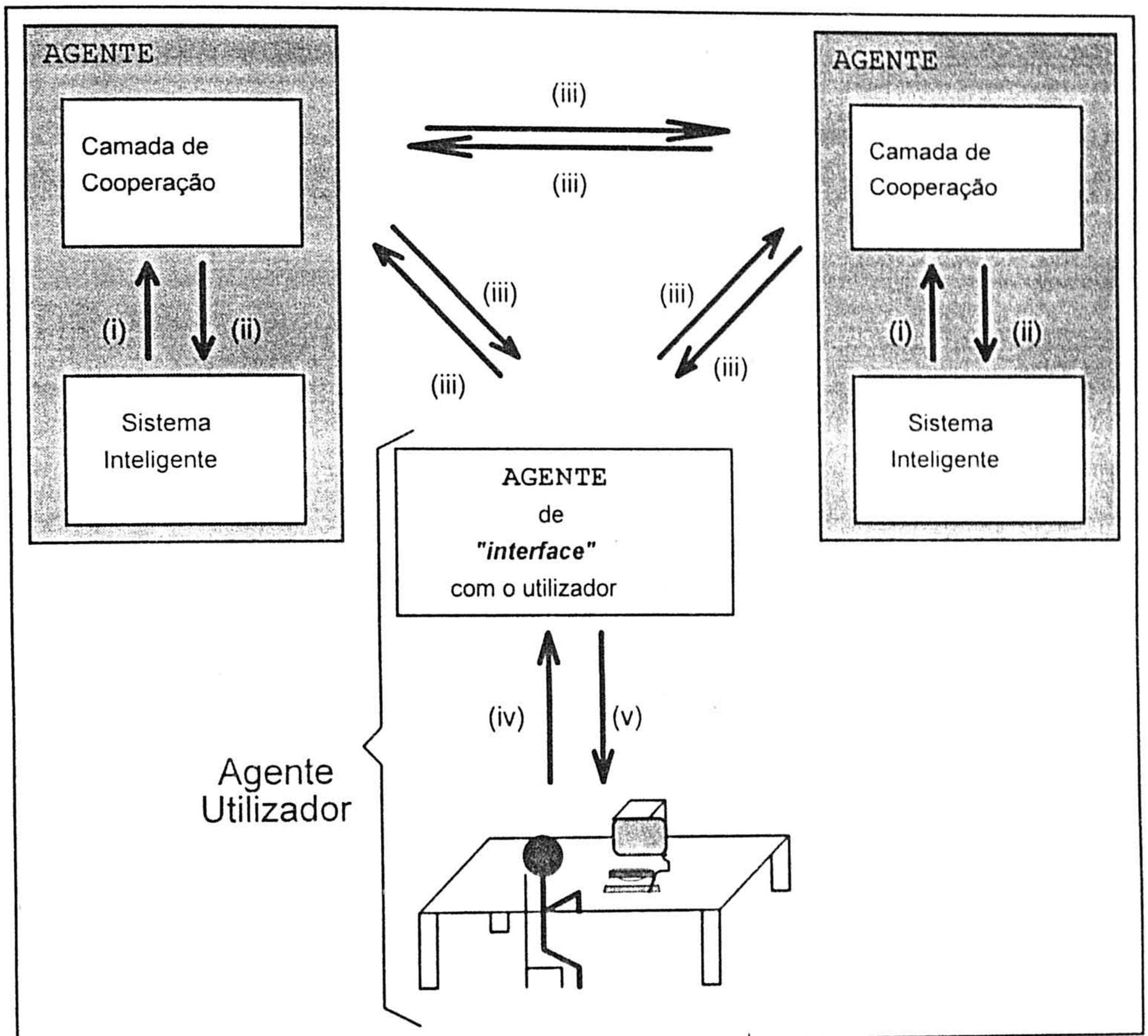
(i) controlar o sistema inteligente a ela acoplado,

(ii) cooperar com os demais agentes da comunidade, e

(iii) coordenar a sua actividade local com a actividade dos demais agentes da comunidade.

Para que estas actividades possam ser executadas, a camada de cooperação comunica com o seu sistema inteligente e com as camadas de cooperação dos outros agentes com os quais possam ocorrer interacções. Esta comunicação é efectuada através da recepção e do envio de mensagens.

Ao conjunto de programas que permitem a comunicação unidireccional do conhecimento entre dois nós é denominada de "*interface*" entre um nó e outro. Cinco diferente "*interfaces*" foram criados entre os diversos nós do sistema multiagente [fig. A3]:



[fig. A3] "*Interfaces*" entre os nós do sistema multiagente

(i) o sistema inteligente e a camada de cooperação que o controla,
(ii) a camada de cooperação de um qualquer agente e o sistema inteligente a ela vinculada,
(iii) duas quaisquer camadas de cooperação de agentes gerados pelo "*UPShell*",
(iv) o utilizador e o agente específico de "*interface* com o utilizador", e
(v) o agente "*interface* com o utilizador" e o utilizador, através do qual o utilizador recebe informações do sistema multiagente ou pedidos de cooperação.

Cada "*interface*" entre dois nós, permite a transferência de conhecimento do primeiro para o segundo nó, sendo o primeiro, o produtor da mensagem, o módulo emissor, e o segundo, que recebe, filtra e analisa as mensagens é chamado de módulo receptor.

A referência [10] contém, por completo, a linguagem para cada um dos "*interfaces*", contudo apresenta-se a seguir alguns exemplos da mesma, utilizando como referência a seguinte estrutura geral:

mensagem(<tipo_de_mensagem>, <id_nós>, <informação>)

A comunicação Intra-agentes

(i) linguagem do "*interface*" - Camada de Cooperação e Sistema Inteligente

a linguagem desenvolvida para este "*interface*" possui os seguintes valores dos não-terminais:

<tipo_de_mensagem> ::= task | info | run | suspend | abort | spawn | next | answer | goHome

<id_nós> :: =

Para este "*interface*" não é necessário a indicação dos nós emissor e receptor, porque o nó emissor é a camada de cooperação do agente e o nó receptor é o sistema inteligente desse mesmo agente.
As mensagens enviadas pela camada de cooperação para o sistema inteligente, a ela vinculado, podem ser dos seguintes tipos:

mensagem(task, Id_Tarefa, Tarefa)
anuncia a tarefa a ser executada,

mensagem(info, Id_Tarefa, Info?)
envia informações necessárias para que a tarefa possa se executada,

mensagem(run, Id_Tarefa)
ordem de início de execução da tarefa,

mensagem(suspend, Id_Tarefa)
ordem de suspender a execução de uma tarefa,

mensagem(abort, Id_Tarefa)
ordem de cancelar uma tarefa, apagando toda a informação relativa a ela,

mensagem(spawn, Id_Tarefa, Id_Tarefa_Cópia) e

mensagem(answer, Id_Tarefa, Id_Subtarefa, Resposta)
caso o sistema inteligente, durante a execução de uma tarefa, encontra uma subtarefa que não consegue resolver, efectua uma pergunta ("*spwan*") à camada de cooperação. Caso a camada de cooperação possua a informação, repassa-a ao sistema inteligente. Caso contrário busca-a no exterior e a repassa.

O sistema inteligente ao receber a mensagem, faz uma cópia do estado da resolução da tarefa, a qual fica agregada ao identificador *"Id_Tarefa_Cópia"* e continua a execução da

tarefa *"Id_Tarefa"*, com a informação recebida em *"Resposta"*. Esta cópia feita pelo sistema inteligente é utilizada, eventualmente, caso se verifique que a informação enviada pela camada de cooperação, da subtarefa, estava desactualizada. Este comportamento possibilita a resolução da tarefa a partir da cópia, usando a nova informação actualizada da subtarefa,

mensagem(next, Id_Tarefa)

efectua uma pesquisa da próxima solução para uma dada tarefa, e

mensagem(goHome)

encerra o funcionamento do sistema inteligente.

(ii) linguagem do "*interface*" - Sistema Inteligente e Camada de Cooperação

mensagem(result, is, Id_Tarefa, Resultado)
mensagem(request, is, Id_Tarefa, Pedido, Id_Subtarefa)

result:

envia o resultado, para a camada de cooperação, de uma tarefa previamente solicitada por ela,

request:

é utilizada pelo sistema inteligente solicitar informações sobre uma subtarefa à camada de cooperação,

is:

significa que a mensagem é proveniente do sistema inteligente (pois a camada de cooperação recebe mensagens de outras camadas de cooperação pertencentes a outros agentes),

Id_tarefa:

identificador agregado à tarefa que esta a ser tratada, e

Resultado, Pedido:

resultado de uma tarefa previamente solicitada ou solicitação de uma subtarefa à camada de cooperação.

A comunicação Interagentes

Os tipos de mensagens enviados por uma camada de cooperação de um agente qualquer para outros agentes, são iguais aos tipos de mensagens recebidos por essa camada de cooperação provenientes de outros agentes.

As mensagens interagentes do sistema multiagente podem ser de dois tipos:

(i) mensagens com conteúdo, que contêm uma descrição do conhecimento que é fornecido ou perguntado, e

(ii) mensagem de controlo e protocolares.

Os vários tipos de mensagens determinam o que os agentes podem realizar com as frases, na linguagem de conteúdo.

De forma não exaustiva, a seguir, apresentamos os grupos de mensagens interagentes, não descrevendo seu formalismo de conteúdo, o qual encontra-se de maneira completa em [10]:

Mensagens de pedido:

query:

efectua uma pergunta a outro agente,

taskRequest:

efectua um pedido para execução de uma dada tarefa, e

nextSolution:

solicita uma solução alternativa para uma questão anteriormente solicitada e atendida.

Mensagem Resposta:

answer:

qualquer resposta refere-se a um determinado pedido efectuado anteriormente, o qual é identificado na mensagem resposta,

Mensagem com Informação voluntária:

volunteer:

estas mensagens permitem o fornecimento de informação a agentes que declaram estar interessados em itens de informação,

Mensagens com Pedido de Explicações:

explain:

estas mensagens permitem que um agente, após ter recebido a resposta a uma pergunta previamente efectuada, peça a explicação da resolução,

Mensagem de Controlo:

control:

permitem controlar o trabalho em curso,

Mensagens Protocolares

protocol:

um tipo de mensagem permite notificar o agente, que faz um pedido para a execução de uma tarefa, da sua aceitação ou rejeição. Outro tipo de mensagem é utilizada durante a negociação para a partilha de tarefas,

Mensagens usadas na Resolução de Conflitos ocorridos na Partilha de Resultados:

my_answer:

os agentes que obtêm resultados de tarefas que podem eventualmente estar em conflito com os resultados de tarefas

idênticas, que outros agentes também estejam a realizar num dado momento, enviam esses resultados para o agente encarregado de resolver conflitos, e

final_answer:

o agente resolvedor de conflitos é o responsável pela obtenção de um resultado final, que deve enviar ao agente destino,

Mensagens para a Visualização da Actividade de um Agente:

post:

o agente de "*interface*" com o utilizador informa, a um determinado agente, que o utilizador pretende visualizar a sua actividade, ou

noPost:

que pretende deixar de a visualizar,

Mensagens para Inspeccionar a Actividade de um Agente:

trace:

o utilizador através do agente de "*interface*", determina quando um dado agente abre o ficheiro para escrita de sua actividade, assim como também determina, de um modo selectivo, que actividade reportar. Esta determinação e a selecção das actividades a serem reportadas, são iniciadas a partir de comandos emitidos pelo utilizador. Como resultado das ordens emitidas pelo utilizador, as mensagens de "*trace*" são enviadas ao agente de "*interface*" com o usuário,

Mensagens para Encerrar a Actividade do Agente:

goHome:

o utilizador, através do agente de "*interface*" com o utilizador, encerra uma sessão de consulta a uma comunidade multiagente, através de comandos "*bye*", "*quit*", ou "*halt*". Como resultado do processamento de qualquer um destes comandos, o agente de "*interface*" com o utilizador envia a cada um dos agentes da comunidade multiagente uma mensagem "*goHome*".

KQML - "*Knowledge Query and Manipulation Language*"

Na cooperação, em uma comunidade de agentes autónomos, os agentes devem ser dotados da capacidade de comunicação. Uma linguagem de comunicação entre agentes ("*Agent Communication Language*") deve ser padronizada e deve ter uma semântica formal para que as diferentes implementaçõ̃es preservem as características essenciais da linguagem.

A "*KQML*" ("*Knowledge Query and Manipulation Language*") [117, 118, 125] é uma linguagem e um protocolo que suporta uma programação em rede, especificamente para sistemas baseados em conhecimentos ou agentes inteligentes. Em termos linguisticos, a "*KQML*" partilha uma:

(i) **sintaxe**, a qual trata de como os símbolos são estruturados,

(ii) **semântica**, a qual trata do significado dos símbolos, e

(iii) **pragmática** a qual trata de como estes símbolos são interpretados,

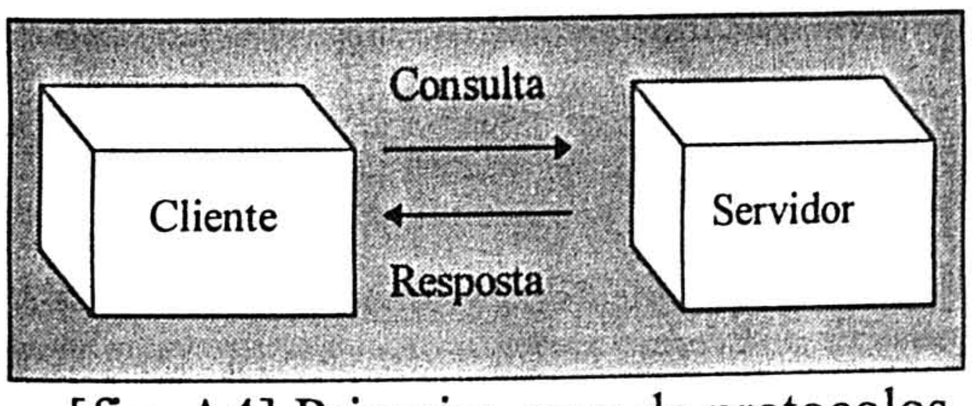
incluindo os participantes da comunicação e a forma de localização dos mesmos.

Dois ou mais agentes podem comunicarem entre si, se eles possuírem uma linguagem de representação que seja comum ou utilizarem linguagens as quais são traduzidas. Assumindo uma linguagem comum ou uma linguagem traduzida, ainda é necessário para a comunicação que os agentes partilhem o conhecimento, para que eles possam interpretar as mensagens trocadas.

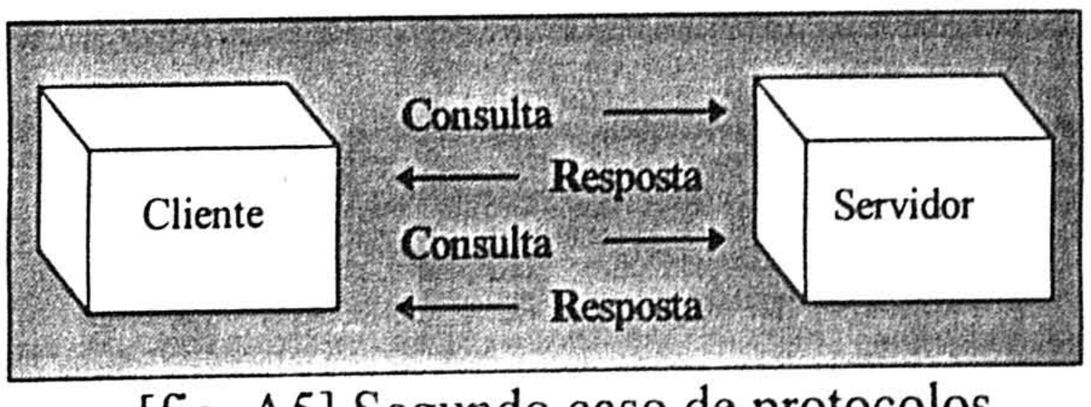
Protocolos "*KQML*"

Existem uma variedade de protocolos de troca de informações entre processos. Existe o simples caso de um processo (um cliente) enviando uma pergunta (consulta) a outro processo (servidor) e esperando por uma resposta [fig. A4]. É comum encontrar este caso em uma inferência do tipo encadeamento para trás, o qual recupera informações de uma fonte remota. Como uma inferência necessita de dados, há a submissão de consultas e a espera pela resposta, antes de executar outra inferência.

Outro caso ocorre quando a resposta do servidor não é completa, mas permite ao cliente solicitar os demais componentes da resposta, um a cada vez [fig. A5]. Um exemplo bastante comum deste tipo de troca, é quando um cliente está consultando uma base de dados relacional ou um servidor, os quais produzem como resposta a consulta uma sequência de instâncias. Muito embora estas trocas requeiram que o servidor mantenha algum estado interno da consulta, cada uma das transações específicas comportam-se idênticas ao primeiro caso. Cada transação é um "envia uma consulta/espera/recebe a resposta". As transações são classificadas como assíncronas, porque as mensagens chegam ao cliente apenas quando elas são esperadas.

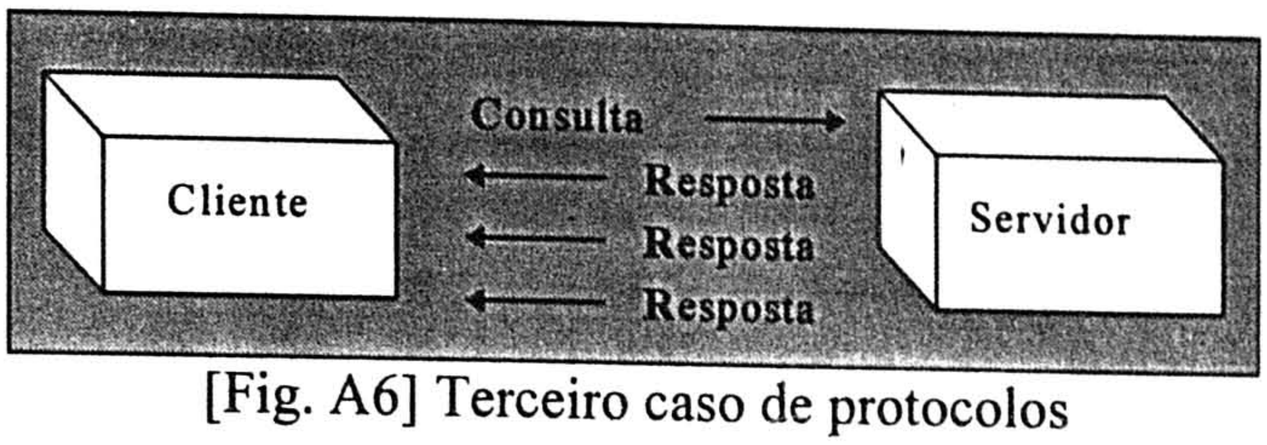


[fig. A4] Primeiro caso de protocolos



[fig. A5] Segundo caso de protocolos

Um terceiro caso, é quando a mensagem é enviada para um específico servidor, cujas respostas chegarão de maneira síncrona ou assíncrona, opcionalmente, associadas com a consulta que eles estão respondendo [fig. A6].



[Fig. A6] Terceiro caso de protocolos

A Linguagem "KQML"

A "*KQML*" suporta os protocolos mencionados, tornando-os parte explicita da linguagem de comunicação. Através da "*KQML*" um agente transmite mensagens compostas em sua própria linguagem de representação, envolvida em uma mensagem "*KQML*".

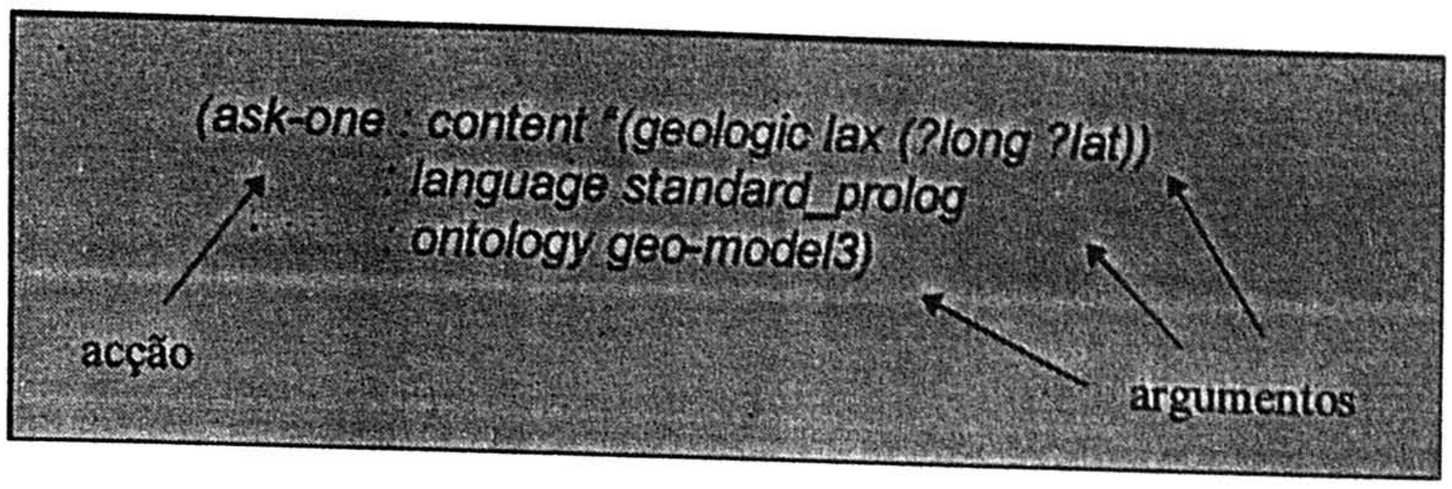
A linguagem "*KQML*" é uma linguagem estruturada em camadas. Ela pode ser vista como sendo dividida em três camadas:

(i) a camada de conteúdo, a qual contém a mensagem na própria linguagem de representação, como caracteres "*ASCII*", uma notação binária ou no formato da troca de conhecimento ("***Knowledge Interchange Format***" [126]). Ela é essencialmente cálculo de predicado de primeira ordem ou linguagem estruturada de obtenção de informações ("***Structured Query Language***" [127]),

(ii) a camada de mensagem compõe a essência da linguagem. Ela determina quais os tipos de interacções. A função primária do nível de mensagem é a de identificar o protocolo de entrega da mensagem, para prover um acto de discurso ("*speech act*")[128] ou uma acção a qual o emissor vincula o conteúdo. A acção significa que o conteúdo é uma assertiva, uma consulta, um comando ou qualquer elemento de um conjunto de acções conhecidas. Devido ao conteúdo da mensagem ser ignorado pela "*KQML*", esta camada inclui também características opcionais as quais descrevem: sua linguagem, a ontologia considerada e algum tipo mais geral de descrição, e

(iii) a camada de comunicação, a qual envolve um conjunto de características que descrevem os parâmetros de mais baixo nível de comunicação, tal como o identificador do destino.

Uma mensagem "*KQML*" consiste de uma acção, seus argumentos associados (conteúdo real da mensagem) e um conjunto de argumentos opcionais os quais permitem tornar independente a sintaxe do conteúdo da linguagem. Uma mensagem representando uma consulta sobre a localização de um particular aeroporto pode ser codificada como apresentada a seguir:



A sintaxe da "*KQML*" é baseada em uma lista de parênteses balanceada. O elemento inicial de uma lista é uma acção e os demais elementos são os argumentos desta acção (pares de palavras chave/valor).

O conjunto de acções "*KQML*" é extensível. Existe um conjunto de acções reservadas as quais têm um significado bem definido. Um agente "*KQML*" pode escolher para manipulação apenas umas poucas acções. Entretanto, uma implementação que opte por uma das acções reservadas, deve ser implementada em um caminho padrão. Uma comunidade de agentes pode escolher para uso, acções adicionais, se os agentes concordarem em suas interpretações e no protocolo associado com cada uma.

Algumas das acções são apresentadas a seguir:

(i) consulta:

evaluate, ask-if, ask-in, ask-one e task-all

(ii) pesquisa multiresposta:

stream-in, stream-all

(iii) respostas:

reply, sorry

(iv) informação geral:

tell, achieve, cancel, untell, unachieve

(v) geração:

standby, ready, next, rest, discard, generator

(vi) definição de capacidades:

advertise, subscribe, monitor, import, export

(vii) redes:

register, unregister, broadcast, route

porém, a "*KQML*" contém acções relativas a aspectos não protocolares da pragmática, tais como:

(i) "*advertise*", a qual permite um agente anunciar que tipo de mensagem assíncrona ele deseja manipular, e

(ii) "*recruit*" a qual pode ser utilizada para encontrar agentes adequados para particulares tipos de mensagens.

Como exemplo, dado dois agentes, agente_**A** e agente_**B**, em um cenário onde o agente_**B** está enviando a seguinte mensagem para o agente_**A**:

```
(advertise
        :language KQML
        :ontology K10
        :content (subscribe  :language KQML
                             :ontology K10
                             :content (stream-about
                                          :language KIF
                                          :ontology motors
                                          :content motor_1)))
```

para a qual, o agente_B responde com:

```
(subscribe
        :reply-with s1
        :language KQML
        :ontology K10
        :content (stream-about
                        :language KIF
                        :ontology motors
                        :content (content motor
                                        :language KIF
                                        :ontology motors
                                        :content motor_1)))
```

o agente_**A,** então, envia um cadeia de mensagens "*tell*" e "*untell*", com informações sobre motor_1:

```
(tell
        :language KIF
        :ontology motors
        :in-reply-to s1
        :content (= (val (torque motor_1) (sim-time 5))
                    . (scalar 12 kgf))

(tell
        :language KIF
        :ontology structure
        :in-reply-to s1
        :content (fastens frame 12 motor_1))

(untell
        :language KIF
        :ontology motors
        :in-reply-to s1
        :content (= (val (torque motor_1) (sim-time 5))
                    (scalar 12 kgf))
```

Anexo C

Propriedades Gerais de Agentes e as Propriedades Específicas para Agentes Deliberativos

De forma geral, para que um agente possa executar uma tarefa, este agente deve ser [129]:

(i) capaz,
(ii) perceptivo,
(iii) bem sucedido,
(iv) reactivo, e
(v) reflexivo.

Um agente é "*capaz*" se seus actuadores são capazes de executar uma tarefa (alterar um espaço de estados). Um agente capaz reflecte as habilidades que seus actuadores possuem de realizar a tarefa. Uma possibilidade para demonstrar a capacidade de realização, é mostrar (comparação) que existe algum agente com os mesmos actuadores que são bem sucedidos, embora ele possa não ter a mesma percepção ou uma mesma função de controlo como a do agente a que estar a ser comparado. A propriedade de ser capaz ignora o facto que um dado agente pode não perceber as relevantes características do ambiente e pode não escolher o comportamento mais adequado.

Um agente é "*perceptivo*" quando possui sensores necessários para determinar como operar seus actuadores para a realização de uma tarefa. A propriedade de percepção habilita um agente a distinguir as importantes características de um ambiente. As importantes características são fortemente determinadas pela tarefa. Diferentes tarefas requerem que o agente responda diferentes características e eventos do ambiente. Por exemplo, se a tarefa é a de abrir um guarda-chuva quando estiver chovendo, a chuva é uma importante característica a ser percebida.

Um agente "*bem sucedido*" é um agente perceptivo com um controlador que possibilite a realização de uma tarefa. Um agente bem sucedido geralmente realiza sua tarefa a partir de todos os estado iniciais.

Existe uma discordância sobre o termo "*reactivo*" quando aplicado a agentes. O "*American Heritage Dictionary*" define reactivo como "responder ou reagir a estímulos". Em Inteligência Artificial, uma definição comum é a de "responder rapidamente e apropriadamente as mudanças do ambiente". Considerando a realização de uma tarefa, o agente deve responder suficientemente rápido e de maneira apropriada as mudanças do ambiente. Um agente que é bem sucedido é então pela definição, suficientemente reactivo.

Um agente é "*reflexivo*" se ele responde apenas a estímulos imediatos, o que causa ambiguidade com o termo reactivo. Um agente com a propriedade de ser reflexivo, não necessita manter qualquer memória.

Um tipo de agente, em particular, de interesse dessa dissertação, e de interesse da comunidade de Inteligência Artificial é o agente "*deliberativo*". Um agente deliberativo possui internamente um modelo do mundo e utiliza-o para:

(i) seleccionar os comportamentos que realizarão as tarefas, e

(ii) raciocinar sobre os efeitos dos comportamentos.

O modelo interno do agente sobre o ambiente, deve providenciar certa funcionalidade básica para que o agente possa raciocinar sobre as consequências de seus comportamentos:

(i) o modelo deve predizer como o comportamento afectará o estado do ambiente,

(ii) modelo deve ser capaz de derivar informação sobre o estado do mundo, a partir de sensores, e

(iii) quanto aos sensores, o modelo também deve ser capaz de predizer qual percepção esperar em um certo futuro estado do mundo.

Adicionalmente ao modelo, o agente necessita avaliar o estado actual do ambiente. A partir da avaliação do estado actual do ambiente, o agente executa projecções para inferir as consequências de potenciais acções.

O resultado de uma deliberação do agente é um plano para execução de uma tarefa. Este plano deve permitir o agente revisá-lo e alterá-lo com novas informações, e possibilitar ao agente questionar a escolha dos comportamentos para a execução de uma tarefa.

O estado interno de um agente deliberativo apresenta seis componentes:

(i) um interpretador de relações. É o modelo dos sensores do agente, o qual indica qual é a percepção que o agente acredita a partir de um certo estado do ambiente,

(ii) um projecto de relações. É o modelo de como o ambiente é alterado em resposta a um dado comportamento,

(iii) um estimado estado do ambiente,

(iv) um plano, o qual desempenha o papel de coordenador e representa as intenções de respostas,

(v) uma tarefa, e

(vi) alguma memória de trabalho.

Um sistema de agentes deliberativos é um sistema multiagente que possui um estado interno deliberativo. O raciocínio imposto sobre o estado do mundo e o planeamento de tarefas são modelados como actualizações do estado interno do agente. O agente selecciona o próximo comportamento para execução através da interpretação de seu plano. Para haver consistência, o plano é restritivo para:

(i) produzir apenas comportamentos que os mecanismos do agente possam aceitar, e

(ii) aceitar todas as percepções que os mecanismos do agente podem gerar.

Devido aos modelos do mundo e das tarefas estarem armazenados no estado (memória) do agente, este pode modificá-los, consistindo na aprendizagem.

As propriedades [fig. A7] de um agente deliberativo depende de seu modelo do mundo, para capacitá-lo a executar sua tarefa:

(i) **predizer**:

um agente prediz se o seu modelo do mundo permite antecipar o resultado de seus comportamentos. Uma "correcta" predição deve predizer todas os possíveis estados que poderiam resultar e não predizer os estados impossíveis,

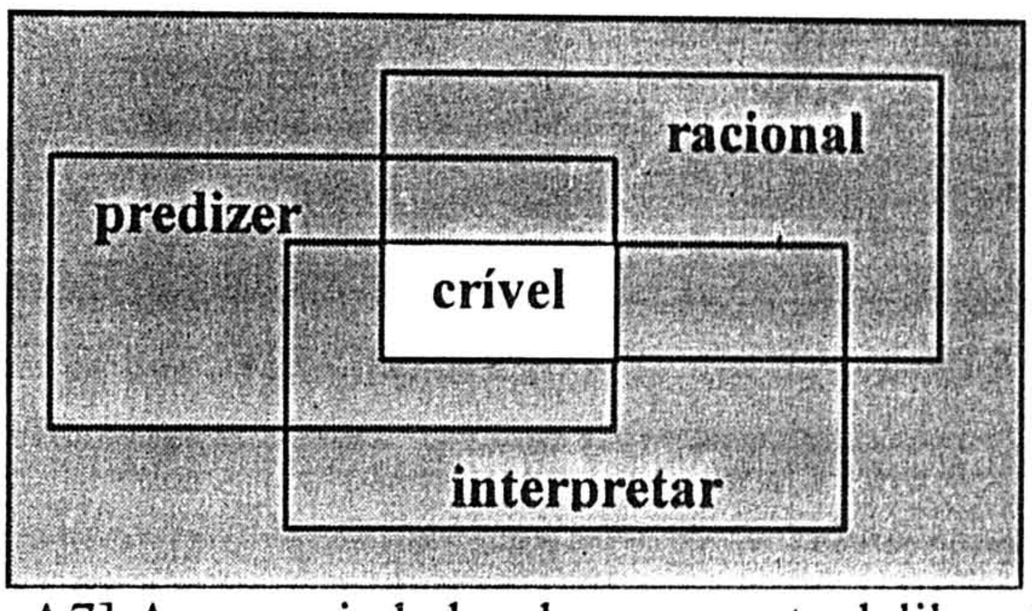
(ii) **interpretar**:

um agente interpreta se o seu modelo de sensores permite uma "correcta" interpretação da percepção por ele recebida. A "correcta" interpretação inclui todos os possíveis pares (estado do mundo, percepção), mas não inclui aquelas que não são possíveis, e

(iii) **ser racional**:

um agente é racional se ele adopta planos que foram por ele preditos que sucederiam, em relação aos planos que foram preditos que não sucederiam. Um agente deliberativo pode predizer o futuro pela aplicação repetida da interpretação do estado corrente do mundo. O agente pode usar sua habilidade de predição para simular a execução de seu plano actual. Se esta simulação conduzir para a execução da tarefa, então é predito para suceder. No planeamento de uma tarefa, o agente revisa seu plano na tentativa de melhorá-lo, podendo, ocasionalmente, adoptá-lo. Um agente é racional se o plano revisado que ele adopta é predito para ser ao menos bem sucedido como o plano anterior.

De forma geral, um agente racional é um agente que decide por planos onde a probabilidade de sucesso foi predita ou a expectativa de utilidade, quando da realização do plano, é alta.



[fig. A7] As propriedades de um agente deliberativo

Um agente é **crível** se ele prediz, interpreta e é racional. Um agente crível não é necessariamente um agente bem sucedido. O agente pode não ser capaz ou não perceber o suficiente para executar a tarefa. Em tal caso, não existindo o plano para realizar a tarefa, o agente, eventualmente poderá gerá-lo, desde que em um tempo que não exceda o início da execução tarefa intencionada.

Anexo D

Arquitecturas de Agentes

(i) Agentes deliberativos

(i.i) Agentes planeadores

O planeamento é essencialmente programação automática. O planeamento é um projecto de um curso de acção que, quando executado, resultará no alcance de um objectivo desejado.

O sistema de planeamento "*STRIPS*" [130] descreve de forma simbólica o mundo, o objectivo desejado e um conjunto de acções com suas pré e pós condições associadas. O sistema procura encontrar, através do "means-end analyses [131]", uma sequência de acções, dentre as do conjunto de acções, que alcance o objectivo desejado.

Duas inovações foram desenvolvidas sobre a versão original do "*STRIPS*": o planeamento hierárquico e o não linear [132, 133].

Os sistemas a seguir, são sistemas de agentes que foram construídos tendo como componente o planeamento:

(i) "*IPEM*" ("*Integrated Planning, Execution and Monitoring*") [134] baseado em planeamento não linear,

(ii) "*AUTODRIVE*" [135], possui agentes planeadores operando em ambientes altamente dinâmicos - simulação de tráfego,

(iii) os "*SOFTBOTS*" [27] os quais podem planear e agir em um ambiente "*UNIX*", e

(iv) "*PHEONIX*" envolve agentes planeadores que operam em domínios de gestão de incêndios florestais [136].

(i.ii) "*IRMA*"

O "*Intelligent Resource-bounded Machine Architecture*" é uma arquitectura baseda em agentes "*Beliefs, Desires and Intentions*", a qual é composta por quatro estruturas simbólicas:

(i) uma biblioteca de planos,

(ii) uma explícita representação das crenças, desejos e intenções,

(iii) um módulo para raciocinar sobre o mundo, e

(iv) um analisador de "*means-ends*" [131], para determinar qual plano pode ser utilizado para alcançar as intenções do agente, um processo de filtragem e um processo de deliberação.

O processo de filtragem é responsável por determinar os potenciais cursos de acção do agente. Esses cursos de acção possuem a propriedade de serem consistentes com as intenções correntes do agente.

A arquitectura "*IRMA*" foi validada em um cenário experimental conhecido como o "*TILEWORLD*" [137]

(i.iii) HOMER

O sistema "*HOMER*" [138]é um sistema de agentes autónomos com habilidades linguisticas, planeamento e execução de acções. Este sistema é um simulador de robô submarino, o qual existe em um mundo marinho de duas dimensões, sobre o qual ele possui um conhecimento parcial.

O sistema "*HOMER*":

(i) comunica-se com o utilizador através de um número limitado de palavras em inglês (+-800),

(ii) as instruções podem conter, moderamente, referencias temporais,

(iii) planeia como executar suas instruções (tipicamente relativas a colecta de informações do mundo marinho),

(iv) executa seus planos, modificando-os conforme requerimentos durante a execução, e

(v) possui uma memória limitada ("*episodic memory*") e, através de seu uso, é capaz de responder questões sobre suas experiências passadas.

(ii) Arquitecturas reactivas

(ii.i) linguagens comportamentais

O trabalho de Brooks [7, 139] propõe três questões:

(i) o comportamento inteligente pode ser gerado sem uma representação explícita do simbolismo utilizado pela Inteligência Artificial?

(ii) o comportamento inteligente pode ser gerado sem explicitar o raciocínio abstracto proposto pela Inteligência Artificial?

(iii) é a inteligência, uma propriedade que emerge de certos sistemas complexos?

Brooks identifica duas ideias chaves:

(i) a inteligência "REAL" está situada no mundo, não em sistemas que utilizam prova de teoremas ou exibem perícia, e

(ii) o comportamento inteligente emerge como resultado da interacção de um agente com seu ambiente, sendo a inteligência não uma propriedade inata e isolada.

Para demonstrar suas ideias, Brooks construiu um número de robôs, baseado em uma arquitectura hierárquica de comportamentos (tarefa, execução). Cada um dos comportamentos disputa com os outros para assumir o controlo sobre o robô. As camadas mais baixas representam as mais primitivas espécies de comportamento (por exemplo evitar obstáculos) e têm a precedência sobre as camadas mais altas da hierarquia.

(ii.ii) Rede de Agente

Foi desenvolvido por Maes [26] uma arquitectura de agentes na qual um agente é definido como um conjunto de módulos competentes. Estes módulos assemelham-se aos da arquitectura de Brooks. Cada módulo é especificado pelo projectista em termos de:

(i) pré e pós condições (como os operadores do "*STRIP*"), e

(ii) um nível de activação, o qual fornece uma indicação de relevância do módulo, em uma particular situação. O módulo de mais alta activação, influenciará o comportamento do agente.

Uma vez especificado, o conjunto de módulos competentes é compilado em uma rede de activação, na qual os módulos são ligados uns aos outros de forma definida pelas pré e pós condições. Por exemplo se módulo **a** tem uma pós condição **x**, e módulo **b** uma pré condição **x**, então os módulos **a** e **b** são conectados por uma ligação "sucessor". Outros tipos de ligações incluem as ligações "predecessoras" e de "conflito". Quando um agente se encontra em actividade, vários módulos tornam-se mais activos em uma dada situação, e podem ser executados. O resultado é um comando para uma unidade actuadora ou talvez o incremento no nível de actividade de um módulo sucessor.

(iii) Arquitecturas hîbridas

(iii.i) PRS

A arquitectura "***Procedural Reasoning System***" [140], a exemplo da arquitectura "*IRMA*" é uma arquitectura de crenças, desejos e intenções. Esta arquitectura inclui uma biblioteca de planos e uma explícita representação simbólica dos estado mentais.

As crenças são factos sobre o mundo exterior ou sobre o estado interno do agente. Estes factos são expressos em lógica de primeira ordem. Os desejos são representados como um sistema de comportamentos (mais que estáticas representações dos estados dos objectivos). A biblioteca de planos do "*PRS*" contém um conjunto parcialmente elaborado de planos denominados de áreas de conhecimento ("*KAs*"), cada uma associada com uma condição de disparo. Esta condição determina quando um "*KA*" é para ser activado. Os "*KAs*" podem ser activados orientados por objectivos ou por dados. Os "*KAs*" também podem ser reactivos, permitindo ao "*PRS*" responder rapidamente as mudanças de seu ambiente. O conjunto dos "*KAs*" activos constituem as intenções dos agentes. Estas várias estruturas de dados são manipuladas por um interpretador, o qual é responsável pela actualização das crenças, chamando os "*KSs*" e executando as acções. O "*PRS*" foi validado em uma simulação de procedimentos de manutenção para o transporte espacial.

(iii.ii) Máquinas de "*Touring*"

A arquitectura híbrida "*TouringMachines*" [141] consiste dos subsistemas de percepção e acção, actuando directamente com o ambiente. Esta arquitectura prevê três camadas de controlo independentes, concorrendo para execução de um processo.

A camada reactiva gera potenciais cursos de acção em resposta a eventos que ocorrem muito rapidamente para que outras camadas possam actuar. Esta camada é implementada com um conjunto de regras (situação, acção), no estilo da arquitectura de Brooks.

A camada de planeamento constrói planos e selecciona acções para alcançar os objectivos do agente. Esta camada consiste de dois componentes:

(i) um planeador, e

(ii) um mecanismo de foco de atenção.

O planeador integra a geração e a execução dos planos. Ele utiliza a biblioteca de planos parcialmente elaborados em conjunto com um mapa topológico do mundo, para construir planos que executarão o objectivo principal do agente.

Por sua vez, a proposta do mecanismo de foco de atenção é limitar a quantidade de informações que o planeador deve tratar. O planeador, desta forma, melhora a sua eficiência, pois as informações irrelevantes apresentadas pelo ambiente são previamente filtradas.

A camada de modelação contém a representação simbólica dos estados mentais de outras entidades do ambiente do agente. Estes modelos são utilizados para identificar e resolver situações de conflito, onde um agente pode não alcançar seus objectivos, como resultado de uma interferência não esperada.

As três camadas são dotadas de comunicação (através de envio de mensagens) e agrupadas em uma estrutura de controlo. A finalidade do controlo é ser o mediador entre as camadas, e em particular para tratar com acções geradoras de conflitos, originárias das diferentes camadas, através de regras de controlo.

(iii.iii) COSY

O "*COSY*" [142] também é um sistema de arquitectura híbrida que inclui elementos tanto do "*PRS*" quanto do "*IRMA*", Ele foi desenvolvido especificamente para ser o ambiente multiagente de teste chamado de "*DASEDIS*" [143].

A arquitectura do "*COSY*" apresenta cinco componentes:

(i) sensores,

(ii) actuadores,

(iii) a comunicação, que é responsável pela envio e recebimento de mensagens,

(iv) a cognição, responsável por mediar as intenções de um agente e suas crenças sobre o mundo, e escolher uma acção apropriada a ser executada. A cognição é a base de conhecimento do agente, contendo suas crenças e três componentes em forma de procedimentos: um roteiro de execução, um protocolo de execução e um componente de raciocínio, decisão e reacção. Os roteiros são receitas ou planos para o alcance de objectivos. Os protocolos representam estruturas de cooperação tal como a rede de contractos ("*Contract Net* [144]"). O raciocínio, a decisão e a reacção são os componentes chaves do "*COSY*", e

(v) a intenção, a qual contém os objectivos a longo prazo, as atitudes, as responsabilidades e os elementos de controlo, que influenciam no raciocínio e na tomada de decisão do componente cognição.

Também figuram como componentes:

(i) uma agenda contendo o número de roteiros activos. Os roteiros podem ser disparados por objectivos ou por dados, e

(ii) um filtro, o qual escolhe entre roteiros competitivos para execução.

Anexo E

Linguagens para Programação de Agentes

(i) Programação orientada à agentes (*Agent-Oriented Programming*)

A linguagem "*AOP*" [110] é baseada no paradigma de uma visão social da computação. A ideia fundamental desta linguagem é a de possibilitar a programação de notações de estados mentais, para a representação das propriedades dos agentes. Os seres humanos utilizam atitudes intelectuais intencionais como um mecanismo de abstracção para descrever humanos, podendo, estas atitudes, serem úteis na programação de máquinas.

Um sistema desenvolvido em "*AOP*" conterá três componentes:

(i) um sistema lógico para definir o estado mental dos agentes,

(ii) uma linguagem de programação interpretada para programar agentes, e

(iii) um processo para compilar agentes em um sistema executável, em baixo nível.

A primeira tentativa de utilização da "*AOP*" foi o sistema "*AGENT0*". O componente lógico deste sistema utiliza uma lógica multimodal, permitindo referência directa ao tempo. A lógica contém três modalidades: crenças, compromissos e habilidades. Como exemplo de uma fórmula da lógica:

$$CAN_a^5\ open(door^8) \Longrightarrow B_b^5\ CAN_a^5\ open(door^8)$$

lê-se: "Se em um tempo *5* o agente *a* assegura que a *porta estará aberta* no tempo *8* então, no tempo *5* o agente *b* acredita que no tempo *5* o agente *a* assegura que a *porta estará aberta* no tempo *8*".

Nesta linguagem, um agente é especificado em termos de um conjunto de competências, um conjunto inicial de crenças e compromissos e um conjunto de regras de compromissos. O componente fundamental que determina como o agente age é o conjunto de regras de compromisso. Cada regra de compromisso contém uma condição de mensagem, uma condição mental e uma acção. Para determinar qual regra deve ser disparada, a condição da mensagem é unida com a mensagem que o agente recebeu e a condição mental é unida com as crenças do agente. Se a regra suceder, então o agente torna-se comprometido com a acção.

As acções podem ser privadas, correspondendo a uma subrotina executada internamente, ou comunicativa, correspondendo ao envio de mensagens. As mensagens estão restritas a:

(i) estimularem ou refrearem a execução de acções, resultando numa modificação de compromissos, e

(ii) informarem, resultando numa alteração das crenças do agente.

(ii) TELESCRIPT (General Magic, Inc.)

A "*Telescript*" [112] é uma linguagem baseada em ambientes, para a construção de sociedades de agentes, desenvolvida pela General Magic, Inc., a qual é tida como a primeira linguagem comercial para agentes.

Esta linguagem encerra dois conceitos básicas:

(i) os lugares, os quais são localizações virtuais que são ocupados por agentes, e

(ii) os agentes, que são os provedores e consumidores de bens em um mercado electrónico. Os agentes são processos de "*software*" os quais possuem mobilidade, sendo capazes de moverem-se de um lugar para o outro através de uma rede. Os agentes são dotados de comunicação, podendo os mesmos residirem em um mesmo local ou em locais diferentes.

Como suporte ao "*TELESCRIPT*", quatro componentes foram desenvolvidos:

(i) a linguagem "*TELESCRIPT*". Esta linguagem é projectada para executar tarefas complexas de comunicação (navegação, transporte, autenticação e controlo de acesso),

(ii) o motor "*TELESCRIPT*", o qual actua como interpretador para a linguagem, mantém os lugares, escalona os agentes para execução, executa a gestão da comunicação e o transporte de agentes e providencia um "*interface*" com outras aplicações,

(iii) o protocolo, o qual providencia a codificação e a decodificação de agentes para serem transportados de um local para outro, e

(iv) as ferramentas de "*software*" para suportar o desenvolvimento de aplicações "*TELESCRIPT*".

Anexo F

Metodologias de Aprendizagem Simbólica Automática: Indução Empírica e Dedução Axiomática

Duas metodologias de aprendizagem simbólica automática são:

(i) Indução Empírica, e
(ii) Dedução Axiomática.

(i) Indução Empírica

A metodologia de aprendizagem automática por indução empírica, utilizada para aprender com o processo de negociação (Capítulo III dessa dissertação), o qual ocorre com agentes que representam as organizações, baseia-se na estratégia de construção de árvores de decisão, como forma de apresentação das regras inferidas. Esta estratégia é utilizada pelo grupo de sistemas de aprendizagem denominado *de "**TDIDT - Top Down Induction Decision Trees**"* [145,155].

Com a sua utilização não restrita a uma área de aplicação, as árvores de decisão induzidas são construídas a partir de um conjunto de exemplos de um fenómeno observado.

Define-se uma árvore de decisão como sendo uma estrutura de dados composta de nós e arcos que unem estes nós, dispostos segundo um critério de ordenação. Os nós são:

(i) nós folhas ou nós de resposta os quais contém um nome de classe, e,
(ii) nós intermediários ou nós de decisão, os quais contém um teste de atributo que, para cada um dos possíveis valores de atributo, existe um ramo para uma outra árvore de decisão.

Os exemplos, obtidos a partir da observação do fenómeno, são representados através de um conjunto de atributos e de seus valores, considerando um dos atributos, a classe do exemplo.

	atr_1	atr_2	...	atr_n (classe)
$observação_1$	$valor_{1,1}$	$valor_{1,2}$	...	$valor_{1,n}$
$observação_2$	$valor_{2,1}$	$valor_{2,2}$	...	$valor_{2,n}$
...	...	...	...	...
$observação_k$	$valor_{k,1}$	$valor_{k,2}$	...	$valor_{k,n}$

O algoritmo geral dos sistemas da família "*TDIDT*" tem como objectivo gerar a menor árvore de decisão que classifique correctamente todos os exemplos. Para que este objectivo seja alcançado, é necessário escolher o atributo mais relevante, i.e. o atributo que melhor particiona os exemplos, segundo o valor da classe. A escolha do atributo, potencialmente melhor é obtido através de uma função denominada de **função de avaliação** [148]. No âmbito dessa dissertação, a **entropia** foi a função de avaliação utilizada.

Entropia:

a entropia é uma grandeza que mede a desordem, tanto de objectos físicos quanto de informação. Quanto maior for a entropia, maior é a desordem do sistema. Um grupo de objectos está tanto mais organizado quanto menos informação externa for preciso para classificá-lo.

Se uma observação pode ser classificada em **n** classes diferentes $\mathbf{c_1}$, $\mathbf{c_2}$, ..., $\mathbf{c_n}$ e a probabilidade de um objecto pertencer à classe $\mathbf{c_i}$ é **p(i)**, então a entropia de classificação do ramo é dada por:

$$E(atr = v_j) = -\sum_{i=1}^{n} p(i)\ log_2 p(i)$$

onde:

atr = v_j: o atributo ***atr*** possui o valor v_j, i.e. a entropria ***E*** do ramo correspondente ao valor v_j do atributo ***atr***

A entropia total de um atributo, é calculada a partir da entropia de cada um dos ramos correspondentes a este atributo:

dados:
(i) a quantidade ***T*** de observações de um certo fenómeno,
(ii) as entropias E_1, E_2, ..., E_n dos ***n*** ramos associados ao atributo escolhido, e
(iii) o número de observações N_1, N_2, ..., N_n de cada ramo,

a entropia de toda a ramificação do atributo ***atr*** é dada por:

$$E(atr) = \sum_{i=1}^{k} E_i * N_i / T$$

Obs: em cada nó da árvore é seleccionado o atributo de menor entropia para continuar a ramificação da árvore.

O processo de escolha de um atributo, utilizando uma determinada função de avaliação, e a ramificação dos exemplos, conforme os valores de um atributo, é repetido para cada nó da árvore, enquanto não se verifique a **condição de parada** do nó. Geralmente a condição de parada é que todos os exemplos do nó, pertençam a mesma classe. Uma vez satisfeita esta condição, o nó torna-se uma folha, nomeada com o valor da classe.

Condições de parada:

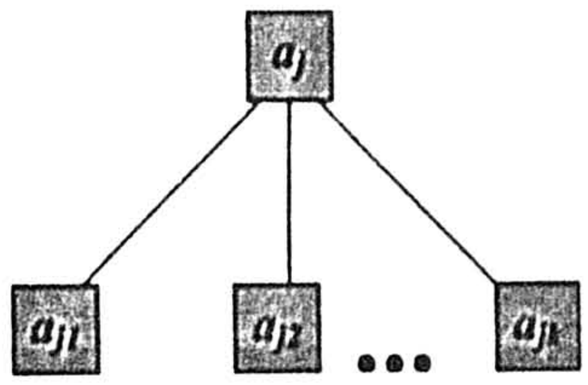
(i) construir a árvore de decisão que classifica exactamente todos os elementos do conjunto de aprendizagem e domínios completos ("*ID3*" [146]),
(ii) decidir pela não expansão da árvore, quando a evidência for insuficiente nos exemplos fornecidos. Este mecanismo de pré-poda de uma árvore de decisão é encontrado no sistema "*C4*" [147] e "*ASSISTANT*" [149]".

Algoritmo geral da família "TDIDT":

dados

um conjunto de exemplos de aprendizagem ***ES***,
uma função de avaliação **a**(**ES**, **atributo**), e
uma condição de parada **s**(**ES**),

se todas as instâncias em **ES** satisfizerem a condição de parada **s**(**ES**),
então retornar o valor da classe,
caso contrário
1. para cada atributo $\mathbf{atr_j}$, determinar o valor da função **a**(**ES**, $\mathbf{atr_j}$),
2. se $\mathbf{atr_j}$ possuir valores $\mathbf{aj_1}$, $\mathbf{aj_2}$, ...,$\mathbf{aj_k}$, criar o nó:



3. particionar os exemplos do conjunto **ES** nos subconjuntos $\mathbf{E_1}$, $\mathbf{E_2}$, ...,$\mathbf{E_k}$ segundo os valores de $\mathbf{a_j}$ na árvore de decisão,
4. aplicar, recursivamente, o algoritmo para cada um dos subconjuntos $\mathbf{E_i}$.

(ii) Dedução Axiomática

A metodologia de aprendizagem automática por dedução axiomática é orientada por exemplos. Estes exemplos são utilizados para apresentar quais novas regras são interessantes serem derivadas a partir do conhecimento de fundo ("*background knowledge*").

Uma das metodologias, para a aprendizagem automática por dedução axiomática, é a de aprendizagem baseada em explicações, enquadrada em problemas denominados de especialização de conceito baseado em teoria ("*theory-based concept specialization*"). A especialização de conceito baseada em teoria é assim denominada, devido a aprendizagem envolver a especialização indutiva de um conceito definido por uma teoria de um domínio.

A seguir são apresentados três métodos [151, 152] para a obtenção de especialização de conceitos, sendo o "*EBG*" e o *mEBG* utilizados para aprender com o processo de negociação:

(i) generalização baseada em explicação ("*EBG - Explanation-based Generalization*"),

(ii) generalização baseada em explicações a partir de múltiplos exemplos (*mEBG - multiple-example Explanation-based Generalization*"), e

(iii) indução sobre as explicações ("*IOE - Induction over explanations*").

Para cada um desses métodos são requeridos:

(i) uma teoria do domínio do assunto,

(ii) um conceito a ser obtido, e

(iii) um ou mais exemplos de treino.

A teoria do domínio define o conceito a ser obtido e o(s) exemplo(s), positivo(s), do conceito a ser deduzido, é(são) especialização(ões) da teoria do domínio.

De forma genérica, o processo de aprendizagem consiste em, repetidamente, aceitar um exemplo de treino e aplicar a teoria do domínio para provar que este exemplo é uma instância do conceito a ser obtido. A partir desta prova e extraindo-se as pré-condições denominadas de "*weakest preconditions*", obtém-se um conceito que, facilmente, pode ser validado. O conceito especializado obtido pode ser utilizado durante a resolução de um problema que necessite que, rapidamente, seja determinado que o(s) exemplo(s) são instâncias da teoria do domínio.

dado

uma teoria do domínio TC a qual define o conceito a ser obtido, e

um conjunto de exemplos positivos de treino de um conceito C, onde C é uma especialização da teoria do domínio TC.

encontrar

uma definição correcta do conceito C.

(i) Generalização Baseada em Explicação

A Generalização Baseada em Explicação forma o conceito a partir de um único exemplo.

Algoritmo "EBG":

Dado um exemplo do conceito a ser aprendido

1. construir uma árvore, que explique porque o exemplo é uma instância do conceito a ser aprendido, e

2. aplicar:

(i) a generalização [150] na árvore resultante, através da substituição das constantes por variáveis, e
(ii) a conjunção das folhas da árvore de explicação, obtendo as "weakest preconditions"

obs: em problemas de especialização de conceito baseado em teoria, as "*weakest preconditions*" formam a definição de um novo conceito, o qual é uma especialização do conceito a ser obtido.

(ii) Múltipla Generalização Baseada em Explicação

Este método difere do método de Generalização Baseada em Explicação, quanto a quantidade de exemplos utilizados para a formação do conceito final.

Algoritmo "mEBG":

Dado dois ou mais exemplos do conceito a ser aprendido:

1. para cada exemplo, construir uma árvore, que explique porque o exemplo é uma instância do conceito a ser aprendido,

2. comparar as árvores obtidas, para encontrar a mais abrangente delas, que é partilhada por todos os exemplos utilizados para a sua construção, e

3. aplicar:

(i) a generalização [150] na árvore resultante do passo **2.**, e
(ii) a conjunção nos termos que figuram nas folhas da árvore resultante, após generalização, extraindo as "weakest preconditions" tal que, a árvore resultante é uma prova válida.

(iii) Indução sobre Explicações

A exemplo do método "*mEBG*", o "*IOE*" também é um método de aprendizagem a partir de múltiplos exemplos. A diferença principal é a do "*IOE*" empregar uma estratégia diferente para a generalização da árvore que explica a maior quantidade dos exemplos utilizados.

Algoritmo "IOE":

Dado dois ou mais exemplos do conceito a ser aprendido:

1. aplicar o método "EBG" para cada exemplo, para produzir o conceito a ser aprendido. Reter as árvores originais que foram utilizadas na produção do conceito,

2. comparar o conceito aprendido em cada uma das árvores produzidas por cada um dos exemplos e providenciar a substituição$###_i$, onde $###_i$ consiste do par $v_j = c_j$,

cada v_j é uma variável no conceito aprendido e cada c_j é uma constante ou um termo do exemplo utilizado ou do domínio da teoria,

3. formar o ### como uma nova substituição para o conceito aprendido, como segue:

(i) faça ### $\leftarrow \varnothing$,

(ii) para cada variável v_j, no conceito obtido, colocar $v_j = c_j$ em ###, onde c_j é obtido como segue:

(ii.i) avaliar se $v_j = c1_j$ ### $###_1$, e

$v_j = c2_j$ ### $###_2$,

(ii.ii) faça cj $\leftarrow$ *gen*($c1_j$, $c2_j$), onde *gen*(c1,c2) é definido como segue:

se

c1 = c2 então *gen*(c1,c2) = c1,

caso contrário

se

c1 = $f(t_{1,1}, t_{1,2},..., t_{1,k})$ e c2 = $f(t_{2,1}, t_{2,2},..., t_{2,k})$

(i.e. termos com o mesmo símbolo de função f),

então

gen(c1,c2) = f(*gen*($t_{1,1}$, $t_{2,1}$), *gen*($t_{1,2}$,$t_{2,2}$),...,

gen($t_{1,k}$,$t_{2,k}$)),

senão

gen(c1,c2) = *v* onde *v* é seleccionada como segue:

se

não existiu nenhuma chamada anterior para *gen* com os mesmos argumentos c1 e c2

então

v é uma nova variável que não figura em nenhum lugar em $###_1$, $###_2$ ou $###_n$. A informação *gen*(c1,c2) = *v* é armazenada em uma tabela para utilização futura.

se

existiu uma prévia chamada para ***gen*** com os mesmos argumentos

então

a tabela de armazenamento é consultada, e a variável previamente gerada é retomada como ***v***.

Anexo G

Sistema Multiagente com a Capacidade de Aprendizagem

"*MALE-Multi-Agent Learning Environment*"

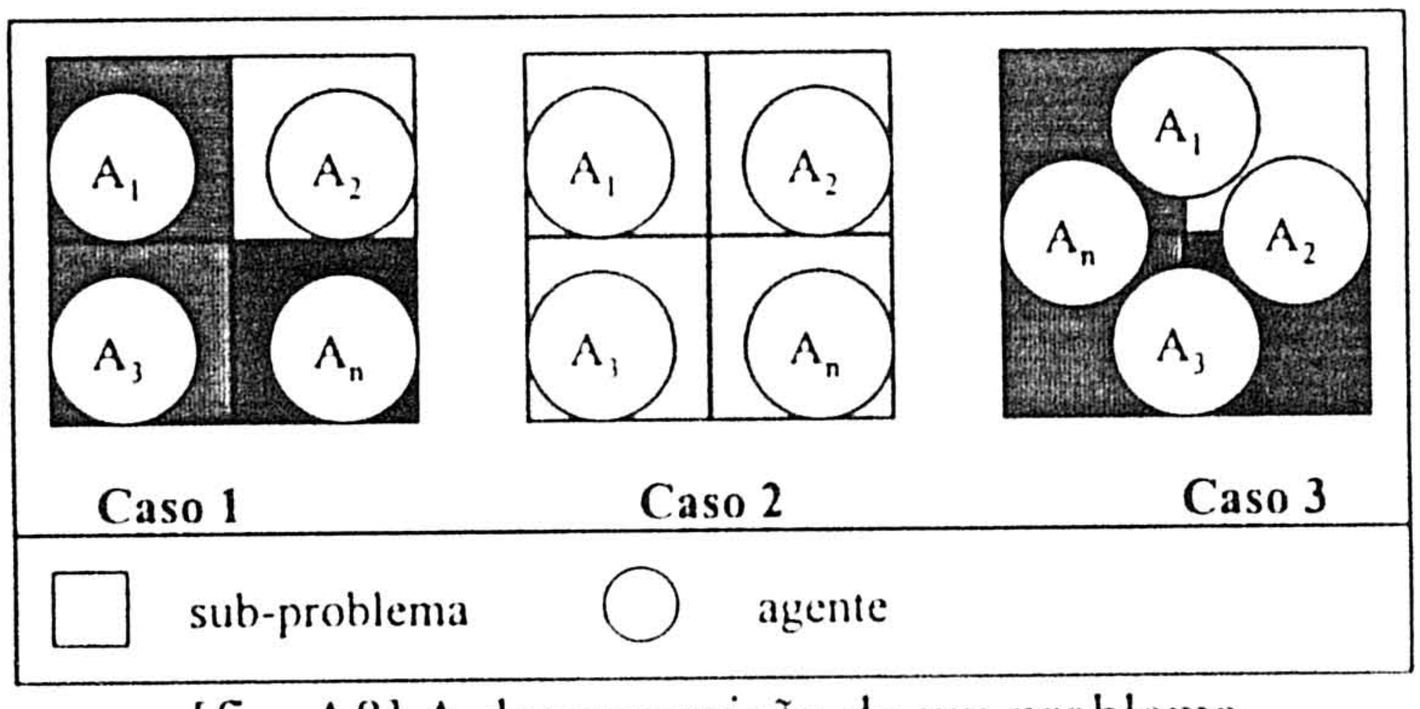
Para operar efectivamente em um mundo real, modelado através de agentes cognitivos, os agentes que o compõe devem ter a capacidade de adaptação a ambientes dinâmicos. O comportamento que permite a adaptação dos agentes, possibilita que os mesmos melhorem os seus desempenhos em tarefas atuais e no manuseio de novas.

Para além da capacidade de adaptação, a cooperação entre agentes também se torna necessária, devido a duas razões:

(i) cada um dos agentes pode apenas ser capaz de inferir uma hipótese parcial, necessitando de cooperação para completá-la, ou

(ii) cada um dos agentes possui limitada experiência e obterá benefícios das experiências com os demais.

A escolha entre as duas razões mencionadas, depende da forma de decomposição do problema a ser solucionado. A decomposição de um problema recai em um dos três casos apresentados [fig. A8] a seguir:



[fig. A8] A decomposição de um problema

(i) caso 1 - múltiplos agentes tratando distintos subproblemas,

(ii) caso 2 - múltiplos agentes tratando subproblemas equivalentes, e

(iii) caso 3 - múltiplos agentes tratando distintos subproblemas, de maneira sobreposta.

O modelo proposto por Sian [52], o qual possibilita a aprendizagem cooperativa, apresenta fundamentalmente três etapas para esta aprendizagem:

(ii) a geração da hipótese inicial, através de um dos agentes,

(ii) a cooperação para obter a concordância da hipótese, e

(iii) a integração da informação aprendida no conjunto de conhecimentos dos agentes envolvidos.

Cada uma dessas etapas apresentam as seguintes questões associadas:

(i) na etapa de geração da hipótese inicial:

(i.i) quando gerar a hipótese. Dentre as possibilidades para a geração de uma hipótese, duas se destacam:

(i.i.i) quando estiver acumulada suficiente evidência, ou

(i.i.ii) quando houver falha para explicar uma observação, e

(i.ii) como gerar a hipótese. Relativamente a esta questão a mesma pode ocorrer:

(i.ii.i) através da segmentação de exemplos, ou

(i.ii.ii) através da aprendizagem a partir de falhas,

(ii) na etapa de cooperação:

(ii.i) a representação do conhecimento. Esta questão, quando modelada com múltiplos agentes, cada um pode possuir uma interpretação diferente da informação, seja devido a diferentes ambientes de actuação, seja pela importância ou pela diferença temporal de ordenação da informação recebida. Para que, cada um dos agentes envolvidos efectivamente cooperem, eles devem ser capazes de utilizar seus conhecimentos de uma forma adequada na hipótese proposta,

(ii.ii) avaliação da hipótese proposta. Uma proposta básica deve ser verificada. Simplesmente indicar que a proposta é inconsistente, é limitar a sua utilidade. Indicar caminhos para torná-la consistente é mais útil, e

(ii.iii) sedimentação da opinião. Esta questão ocorre quando, dado as opiniões dos agentes participantes, surgir a necessidade de decidir sobre a validade da hipótese. Para uma aprendizagem cooperativa, é necessário uma forma efectiva de combinar opiniões, e

(iii) a integração da informação aprendida no conjunto de conhecimentos dos agentes, apresenta a questão sobre as crenças de cada um dos agentes envolvidos na cooperação. Esta questão introduz a revisão de crenças como forma preservadora da verdade.

O modelo proposto, para a construção de sistemas distribuídos, com capacidade de aprendizagem cooperativa e que atenda a algumas das questões mencionadas, assume as seguintes características:

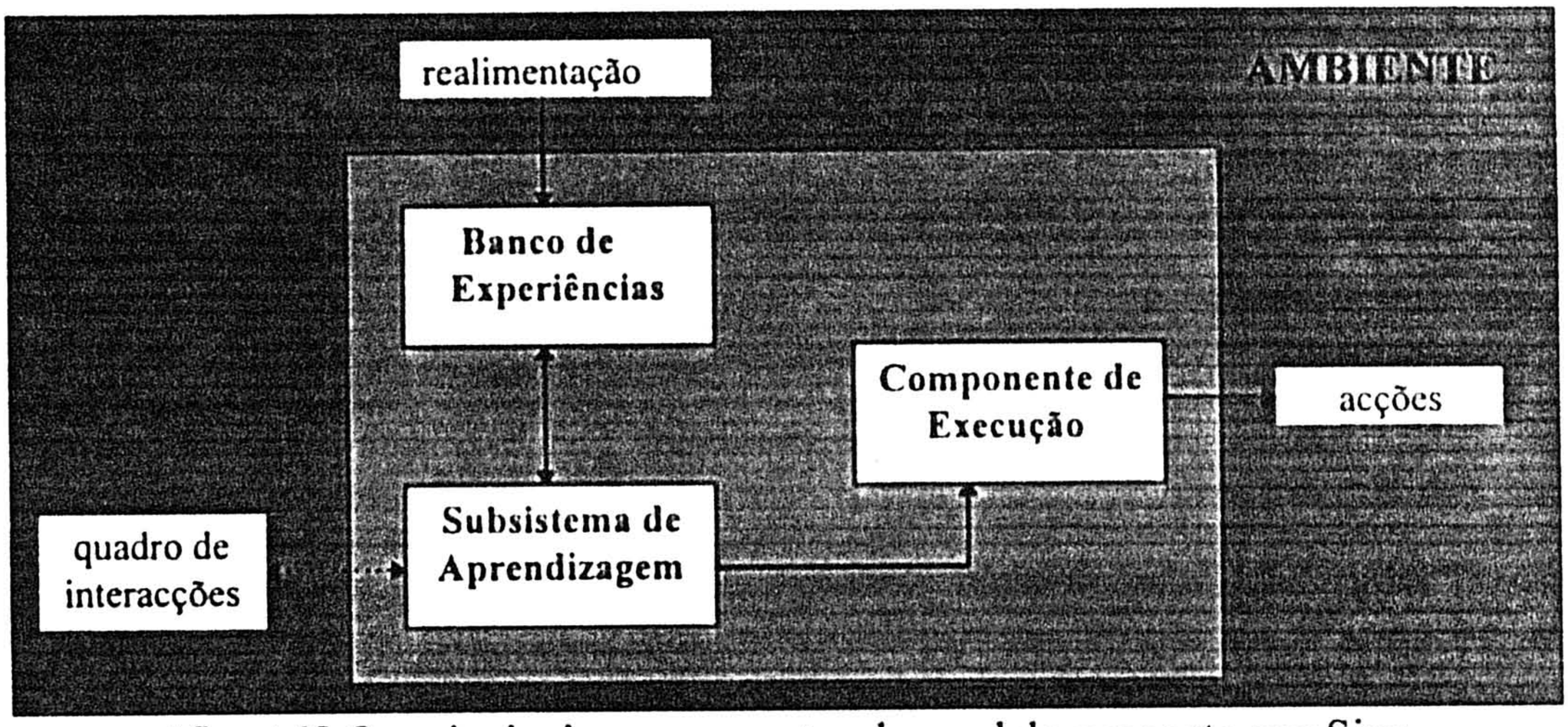
(i) cada agente do modelo é um aprendiz incremental, o qual utiliza de exemplos que não podem ser explicados através do conhecimento actual, como dados para a aprendizagem. Um parâmetro, o qual determina quando foi encontrado um número suficiente de exemplos para que a hipótese seja utilizada como fonte de cooperação, é especificado através de um "*interface*" com o ser humano,

(ii) foi introduzido operadores que permitem que o agente expresse suas opiniões sobre hipóteses e um quadro partilhado no qual cada interacção ocorre,

(iii) foi providenciada uma função de consolidação de opiniões a nível individual, e
(iv) a integração de novo conhecimento é tratada via intervalos de tempo de crenças, baseados em cálculos de eventos.

Componentes do modelo

Os principais componentes do modelo [fig. A9] são:



[fig. A9] Os principais componentes do modelo proposto por Sian

(i) o componente de execução. Este componente consiste basicamente das mesmas funções disponíveis em um agente baseado em Inteligência Artificial Distribuída, i.e. responsável pela execução da solução de um problema e pela comunicação e coordenação requerida,
(ii) o banco de experiências é uma colecção organizada de conhecimentos que o agente recebe quando da realimentação a partir do ambiente, e
(iii) o subsistema de aprendizagem, o qual é composto por duas tarefas principais:

(iii.i) a de decisão sobre o estado das experiências, e
(iii.ii) a de organização do banco de experiências e de execução da aprendizagem, coordenada via o quadro de interacções.

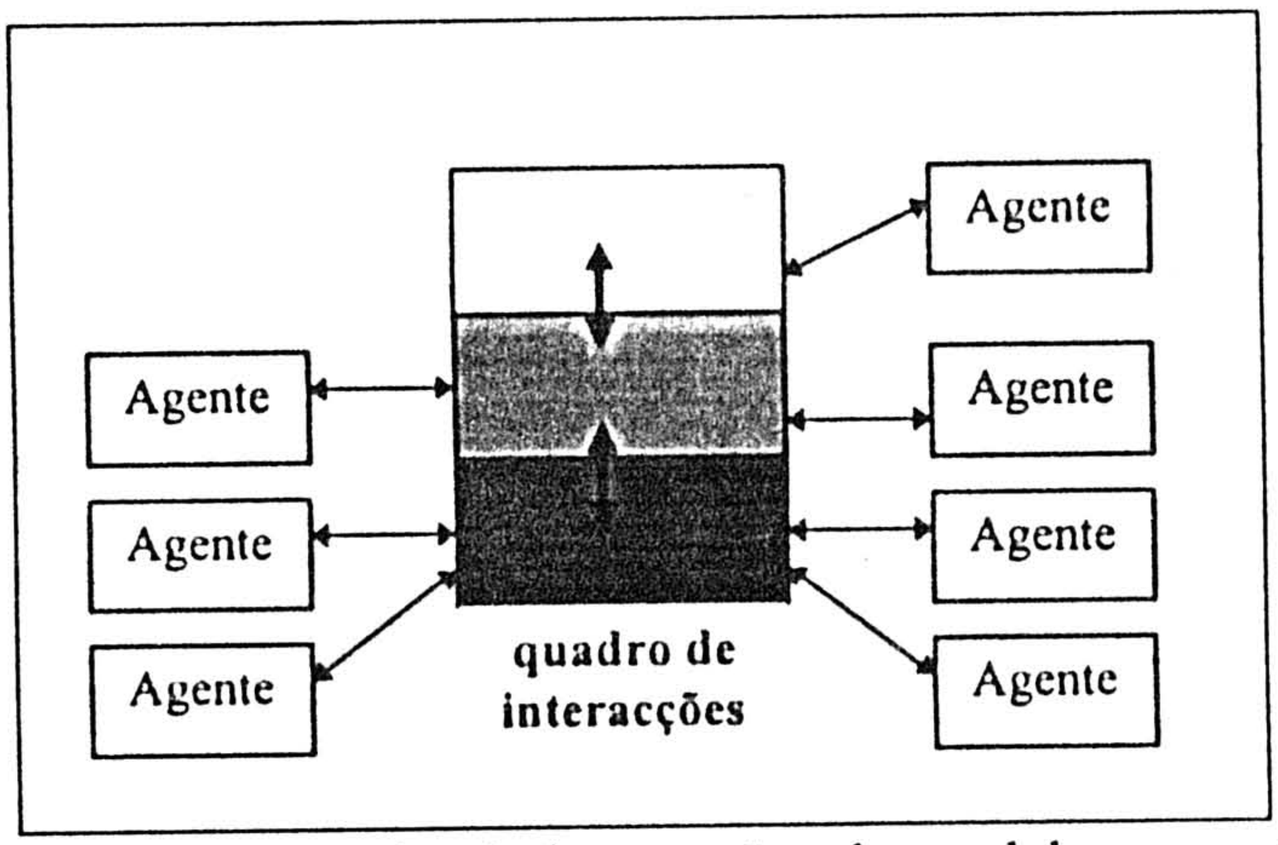
A execução da aprendizagem, utiliza-se da aprendizagem por experiência. Cada item de experiência, que o agente recebe, pode servir para uma das seguintes propostas:

(i) se o exemplo é explicado pelo conhecimento actual do agente, então sua utilidade reside para incrementar a credibilidade do conhecimento, e
(ii) caso contrário, então o conhecimento actua como um exemplo para a aprendizagem incremental. O objectivo do

aprendiz é, de maneira incremental, construir regras que cubram os exemplos não explicados.

O quadro de interacções

O quadro de interacções [fig. A10], baseado no paradigma do quadro-negro [22], é um mecanismo através do qual os agentes podem interagir. Ele consiste de uma estrutura de múltiplos níveis na qual vários agentes são conectados. Os agentes que ocuparem os níveis mais altos, são capazes de inspeccionar e modificar informações dos níveis mais baixos. O quadro de interacções dispõe a notificação para os agentes, quando eles têm que responder a uma proposta ou notifica a proposta quando todos os agentes tiverem respondidos.



[fig. A10] O quadro de interacções do modelo proposto

Linguagem de comunicação interagentes

A linguagem de comunicação interagentes consiste de nove operadores, os quais permitem a comunicação no domínio das hipóteses:

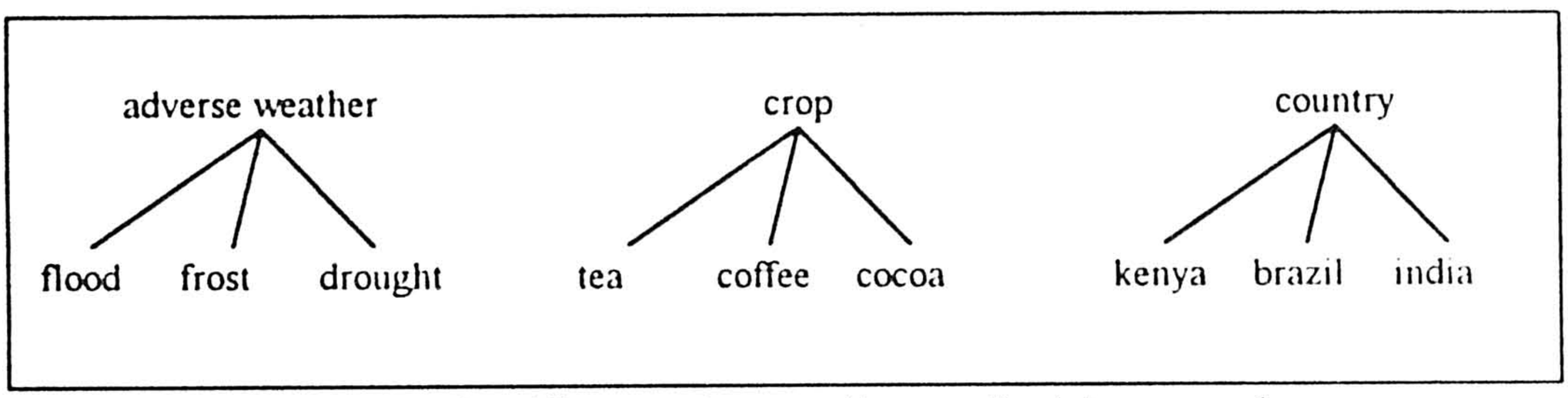
(i) *ASSERT*
assertiva não modificável. Utilizada pelos gestores aos agentes de mais baixo nível,
(ii) *PROPOSE*
propõe uma nova hipótese,
(iii) *MODIFY*
obtém uma versão modificada de uma hipótese previamente proposta,
(iv) *AGREE*
altera o estado, de "propõe" para "concorda", de uma hipótese,
(v) *DISAGREE*
indica a discordância de uma hipótese,
(vi) *NOOPINION*
não expressa uma opinião em relação a uma hipótese,
(vii) *CONFIRM*
indica a confirmação da hipótese,

(viii) *ACCEPT*
causa a aceitação de uma hipótese previamente concordada, e
(ix) *WITHDRAW*
retira a proposta.

Exemplo

Condiremos três agentes trabalhando no domínio do comércio de mercadorias. Vamos assumir que este exemplo, quanto ao tipo de decomposição do problema, é do tipo "caso 2" , i.e. os agentes estão tratando de problemas equivalentes. A aprendizagem é cooperativa para a consolidação da experiência.

Os três agentes negociam com mercadorias na América, África e Ásia. Cada agente trata do mesmo problema, porém recebendo dados de diferentes regiões. O objectivo dos agentes é predizer a variação dos preços, dada as condições predominantes. Todos os agentes têm o conhecimento de uma classificação [fig. A11], disponível como conhecimento comum.



[fig. A11] Classificação das condições climática e produtos

Como complemento, cada agente possui os seguintes conhecimentos:

Agente1	major_producer(kenya, coffee)
	major_producer(kenya, tea)
Agente2	major_producer(brazil, coffee)
	major_producer(brazil, cocoa)
Agente3	major_producer(índia, tea)

É assumido que os agentes ao ingressaram no comércio de mercadorias, em um período de tempo, eles possuem os seguintes exemplos não explicados, armazenados no banco de experiências e construíram as seguintes generalizações a partir desses exemplos:

$Agent_1$	weather(kenya, drought), price(tea, rising) [4 instances]
	weather(kenya, drought), price(cocoa, steady) [4 instances]
	weather(kenya, frost), price(coffee, rising) [2 instances]
Gen:	weather(kenya, adverse), and major_producer(kenya, crop)
	### price(crop, rising)
$Agent_2$	weather(brazil, frost), price(coffee, rising) [4 instances]
	weather(brazil, flood), price(cocoa, rising) [1 instance]
Gen:	weather(brazil, adverse)
	### price(crop, rising)

BIBLIOTECA F.E. PORTO

Agent$_3$ weather(índia, flood), price(tea, rising) [6 instances]

Gen: weather(índia, flood)
price(tea, rising)

Tomando como referencia a figura [fig. A12] assume-se que o Agent$_3$ inicia a interacção com sua hipótese P_1. Os outros agentes devem então responder a hipótese formulada P_1.

Muito embora o Agent$_2$ não tenha provas para confirmar P_1, a generalização é construída a partir de seus dados experimentais, totalmente submetida a hipótese proposta. Entretanto, o Agent$_2$ propõe sua hipótese como uma modificação (neste caso uma generalização) para a proposta P_1 do Agent$_3$, e indica o seu nível de confiança nela. Estas acções agem, ambas, como uma resposta para o Agent$_3$ e como uma nova proposta de uma hipótese P_2, para a qual os outros agentes devem responder.

O Agent$_1$ primeiro responde sobre P_1 para o Agent$_2$, cuja generalização foi construída a partir de seus exemplos, no entanto o Agent$_1$ indica uma alteração, apresentando sua hipótese P_3. Uma vez que todos os agentes responderam sobre P_1, o nível de credibilidade do resultado é calculado utilizando uma função de validação "*Eval()*" [52].

O papel da função de validação é apresentar uma efectiva credibilidade baseada nas respostas recebidas dos agentes. Uma resposta individual pode consistir de "*CONFIRM*", "*NOOPINION*", "*MODIFY*" ou "*DISAGREE*". As demais que não "*NOOPINION*", contém uma métrica de segurança da resposta (a qual é baseada nas experiências obtidas de uma resposta particular), e a "*MODIFY* " também contém uma métrica de similaridade entre a proposta e a hipótese modificada.
Sendo assim, as respostas são da forma:

CONFIRM(H,S)| NOOPINION(H)| MODIFY(H,S,M) | DISAGREE(H,S)

onde: H = a hipótese que está sendo validada,

0< S =<1 é a confiança da resposta e 0< M =<1 é a medida de similaridade.

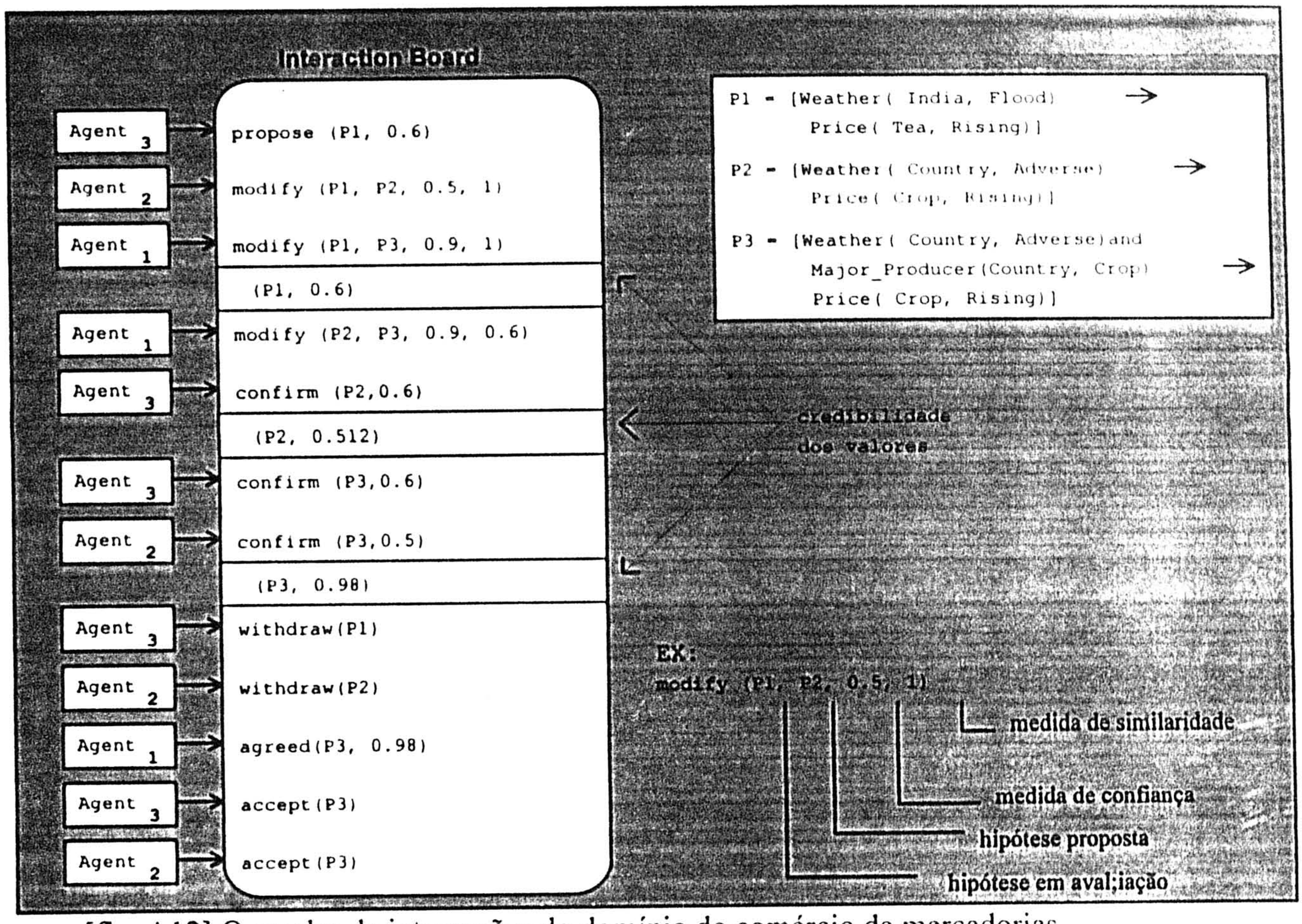
Desde que a hipótese P_1 tenha duas modificações a ela vinculada, o Agent$_3$ deve aguardar até que elas tenham sido validadas, antes de decidir o estado de P_1 através do "*AGREE*" ou "*WITHDRAW*".

O Agent$_1$ responde para o Agent$_2$ sobre a P_2, indicando sua hipótese P_3 como uma modificação, embora neste caso, P_3 seja uma especialização de P_2. Como os dados do Agent$_3$ são totalmente explicados por P_2, ele simplesmente confirma-a, podendo P_2 ser validada.

O Agent$_2$ e o Agent$_3$ confirmam a hipótese P_3 do Agent$_1$, desde que esta esteja consistente com seus dados. O resultado da validação de P_3, obteve o mais alto valor, sendo retiradas as outras duas hipóteses associadas e P_3 é aceita pelos outros agentes.

É importante observar o que os agentes ganharam com esta experiência. A hipótese do Agent$_3$ foi muito específica e a hipótese aceita é potencialmente mais útil para ele. A hipótese do Agent$_2$, embora correcta com respeito a seus dados, foi muito geral. A

hipótese aceita corrige este defeito. O Agent$_1$, cuja hipótese foi aceita, foi capaz de mostrar sua confiança nela, uma vez que os outros agentes confirmaram as evidências.



[fig. A12] O quadro de interacções do domínio do comércio de mercadorias

Anexo H

Metodologia "*ARIS*"

Os sistemas de informação de negócios, suportados por computador, providenciam um veículo para interligar conceitos de aplicações de negócios com tecnologia da informação.

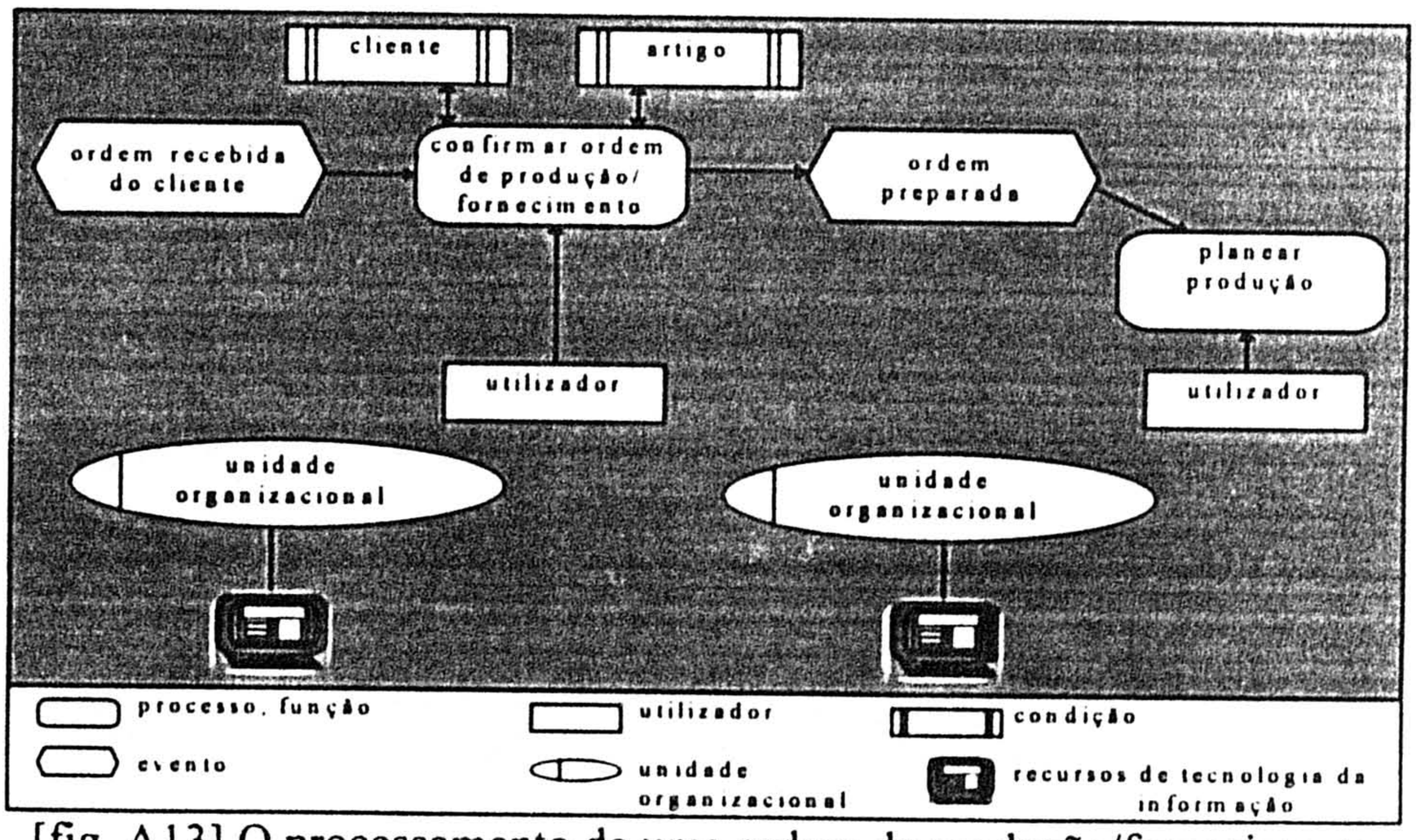
Em sistemas de negócio, uma distinção é frequentemente feita entre sistemas de gestão, de escalonamento de tarefas, de planeamento e de controlo, baseada no tipo de processo que envolve o negócio.

A arquitectura de sistemas integrados de informação "*ARchitecture of Integrated Information Systems - ARIS* " [83] baseia-se no conceito de integração (redução de redundâncias, coesão e acoplamento por dados, por procedimentos e por tempo) o qual suporta os processos de negócios existentes em uma organização.

O primeiro passo da arquitectura "*ARIS*" compreende o desenvolvimento de um modelo para os processos de negócios. Devido estes processos serem altamente complexos, o modelo é construído particionando o negócio em diferentes visões. Como decorrência desta divisão, é possível descrever cada uma das visões através de metodologias especializadas, sem haver a necessidade de incorporar, em um primeiro momento, as correspondentes relações entre as visões. Por fim, após cada uma das descrições, as relações entre as visões são restabelecidas. Adicionalmente a estas divisões, o segundo passo da arquitectura envolve conceituar os diferentes níveis de descrição.

Descrições das visões

A figura [fig. A13] a seguir apresenta a instância de um dos processos de negócio "o processamento de uma ordem de produção/fornecimento", a qual ilustra o conceito das visões.



[fig. A13] O processamento de uma ordem de produção/fornecimento

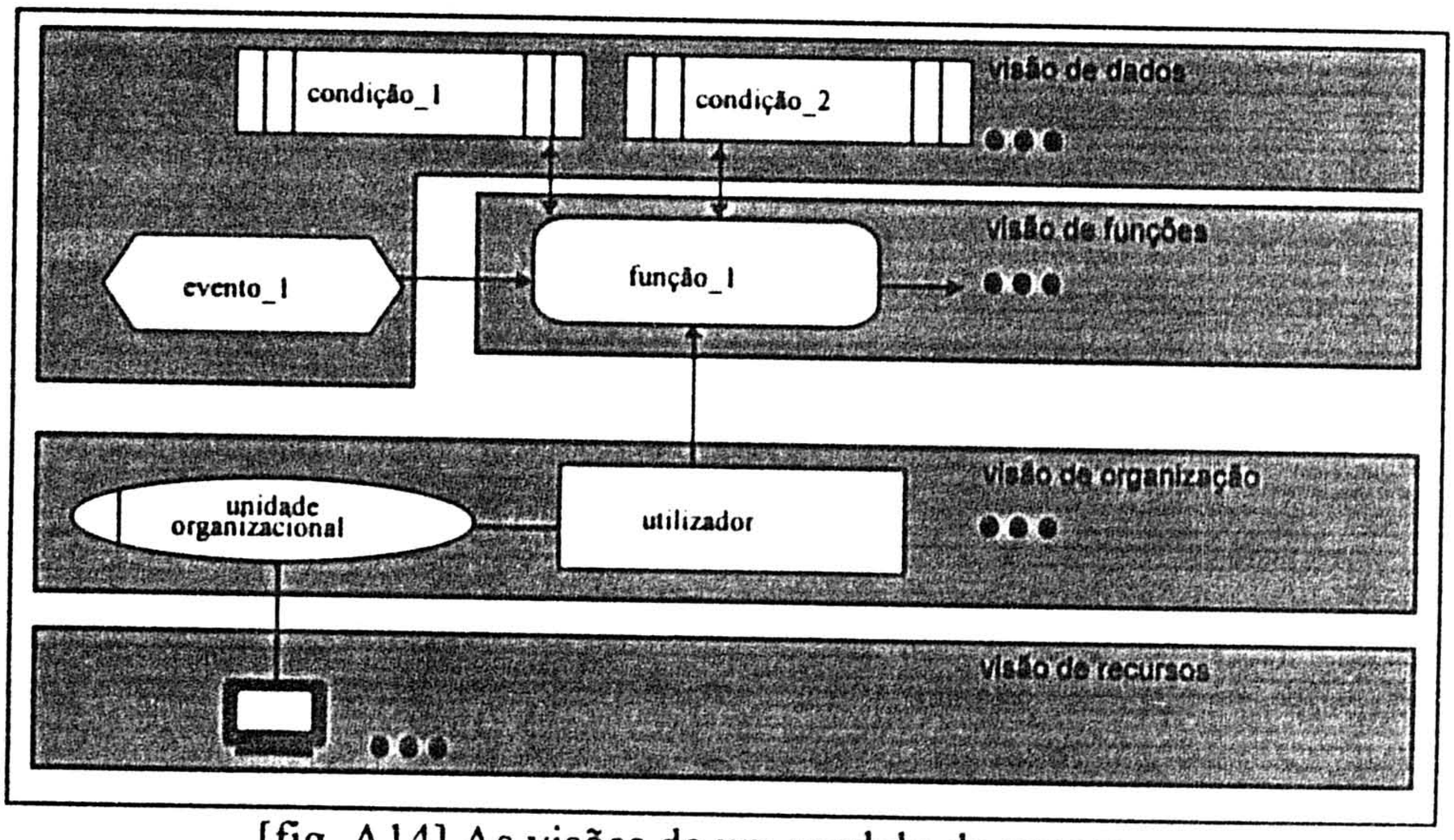
Neste exemplo, o processo de uma ordem de produção/fornecimento é disparado pelo evento "ordem recebida do cliente", o qual por sua vez inicia o processo de confirmar "ordem de produção/fornecimento". As descrições de condições para as operações relevantes do ambiente são necessárias para processar a operação (neste exemplo são as condições que afectam o cliente e os produtos contidos na ordem).
As condições podem ser alteradas durante o processamento do fluxo, por exemplo, se não houver matéria-prima disponível.

Nesta instância, um evento é processado por um ser humano. Este ser humano é designado para um departamento, o qual está equipado com recursos de tecnologia da informação (computador, acesso a base dados, etc.). A confirmação de uma ordem de produção é o resultado do processamento de um fluxo de trabalho. O evento de confirmação de uma ordem de produção, pode disparar adicionais operações, tal como o planejamento de produção. As descrições das condições, bem como os recursos humanos e técnicos são necessários para processar estas operações.

Os componentes e suas relações, para serem descritos em um processo de negócio suportado por computador, incluem:

(i) os processos,
(ii) as actividades,
(iii) os eventos,
(iv) as condições,
(v) os usuários,
(vi) as unidades organizacionais, e
(vii) os recursos da tecnologia da informação.

Se considerado todos os efeitos em todos os elementos de um processo, quando do projecto de um sistema de informação, esta atitude aumentaria a complexidade do projecto do processo. Para reduzir a complexidade, o modelo é dividido em visões individuais [fig. A14], as quais representam aspectos discretos do projecto e podem ser manuseados de maneira independente, o que simplifica a tarefa.



[fig. A14] As visões de um modelo de processos

Os eventos, são objectos de informação os quais são representados por dados. Os atributos de condição tal como o estado do cliente ou estado do artigo, também são

representados por dados. Então, os eventos e as condições formam a "**visão dos dados**".

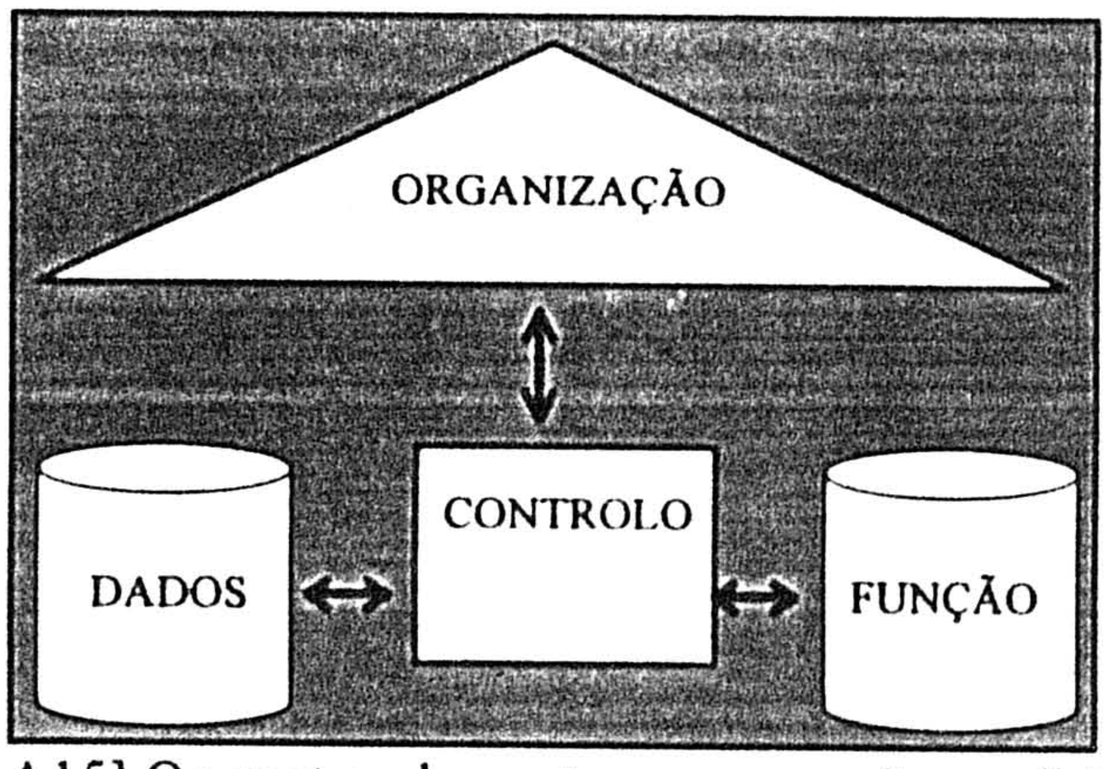
As funções a serem executadas e suas relações, constituem um "**visão de função**". Esta visão contém a descrição individual, a enumeração de individuais subfunções e as relações que existem entre funções.

Devido as estreitas relações existentes, os componentes "utilizadores" e "unidades organizacionais", eles estão consolidados em uma descrição de visão a "**visão da organização**".

Os componentes da tecnologia da informação, constituem a "**visão dos recursos**".

Pelo particionar de um problema inicial em visões individuais, a complexidade é reduzida, porém como as relações entre os componentes, não foram estabelecidas até o momento, uma "**visão de controlo**" é introduzida para desempenhar este papel. A visão de controlo é um componente essencial do "*ARIS*" que se destingue das demais propostas de arquitecturas neste domínio. Através da introdução da visão de controlo na arquitectura, é possível reter as relações entre as visões. A subsequente declaração das entradas que estabelecem o relacionamento entre as visões, torna possível a sistematização de todas as relações.

De maneira geral, o modelo resulta em quatro elementos que compõem o "*ARIS*" [fig. A15].

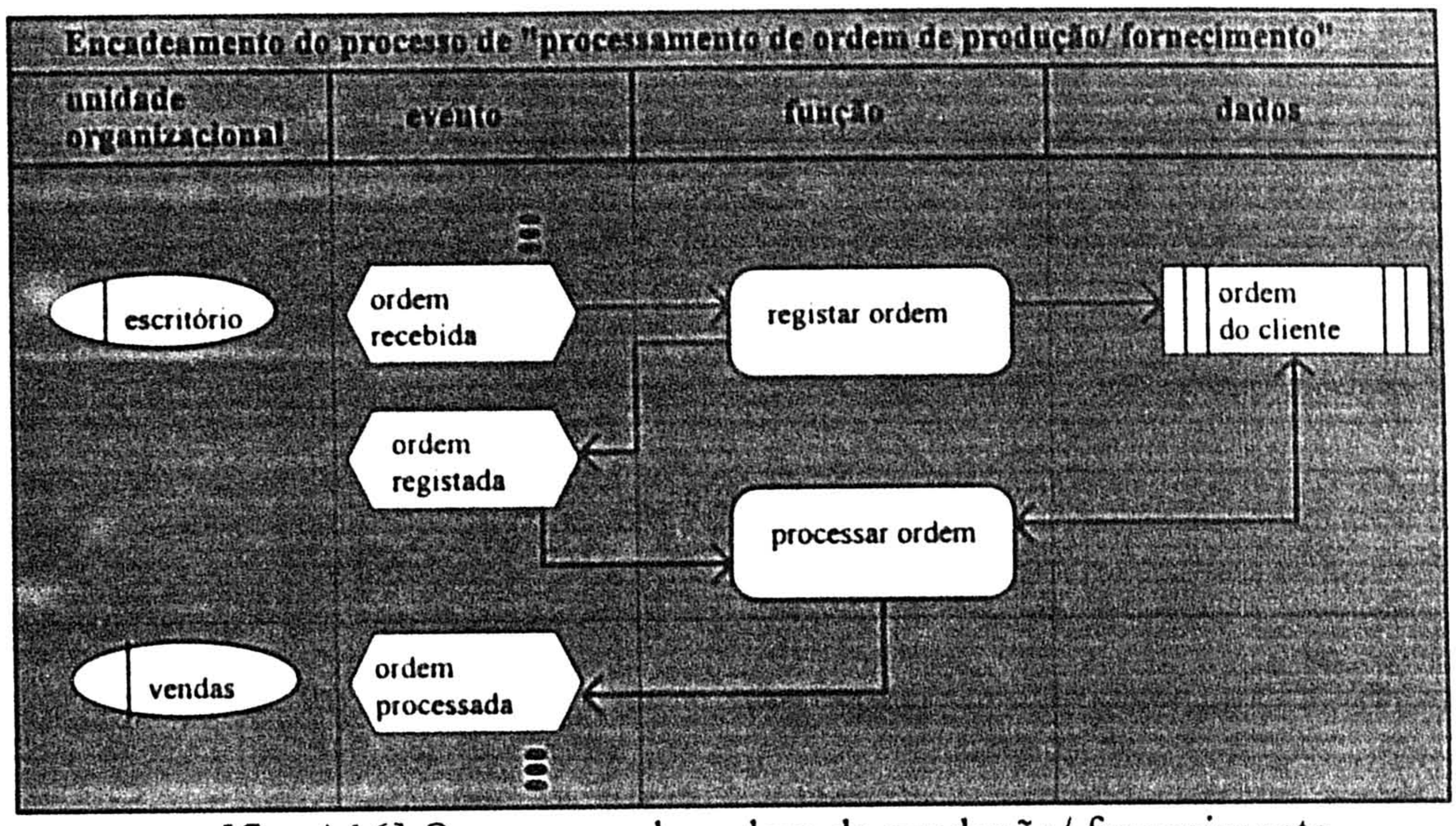


[fig. A15] Os quatro elementos que compõem o "*ARIS*"

Para que estes quatro elementos estejam integrados, na representação de um processo, o diagrama de encadeamento de processos "*Process Chain Diagram*" é utilizado. Estes diagramas providenciam uma representação condensada e contém todas as visões descritivas do "*ARIS*":

(i) funções,
(ii) organização,
(iii) dados, e
(iv) as interacções.

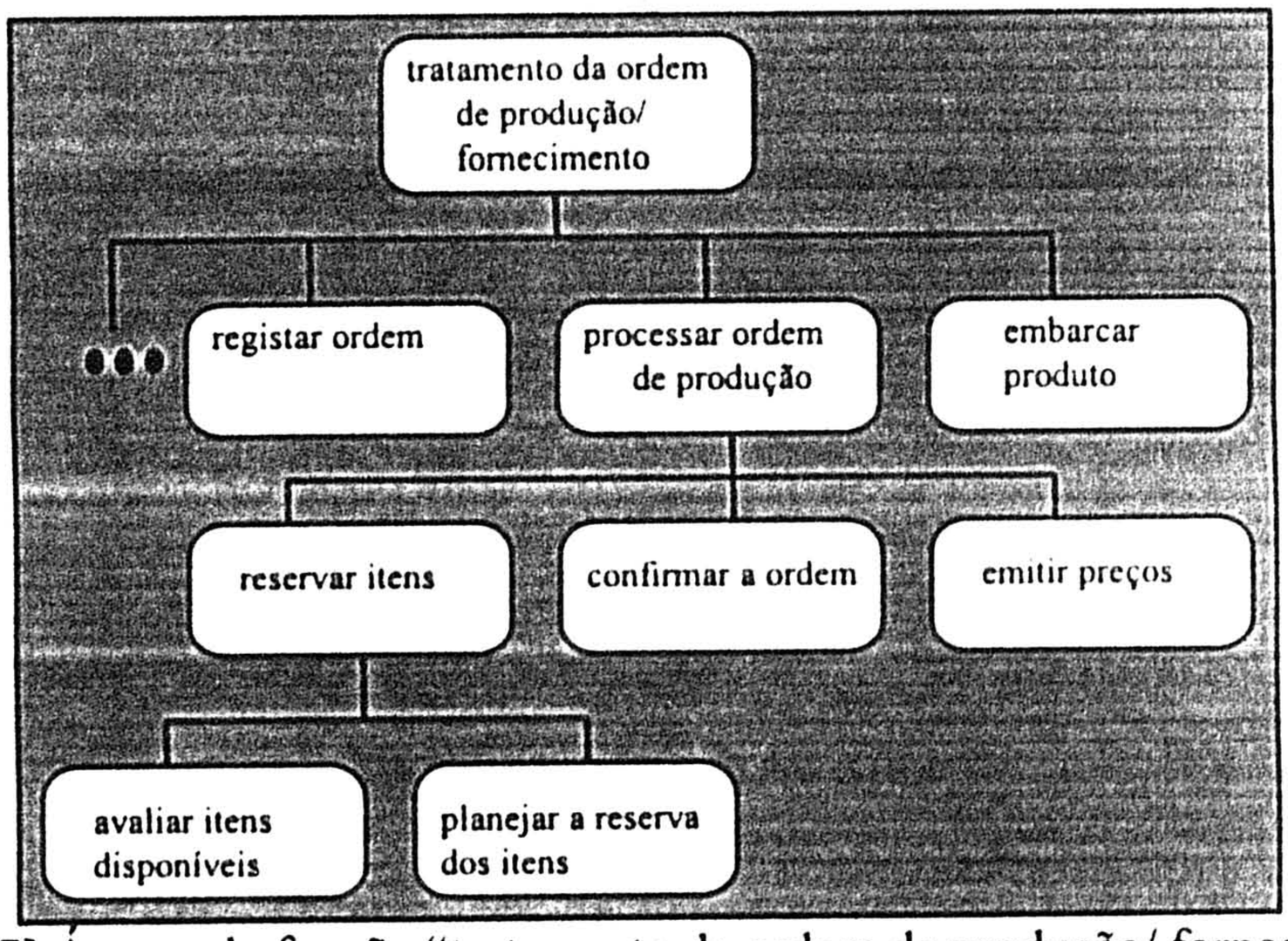
Como exemplo, o processo parcial para atender uma ordem de produção/ fornecimento é apresentado [fig. A16] a seguir.



[fig. A16] O processo de ordem de produção/ fornecimento

A primeira coluna do diagrama, apresenta a unidade funcional responsável pela(s) função(ões) de um processo. A segunda e a terceira coluna apresentam, respectivamente, os eventos (que iniciam uma função e os eventos por esta gerados) e as funções de um processo. A quarta coluna apresenta os dados, de entrada e de saída, processados pelas funções. As interdependências (arcos orientados) ilustram as ligações entre eventos, funções e dados.

As funções criam ou modificam objectos. Uma função complexa pode ser particionada em subfunções, ocorrendo uma redução de complexidade de conceito e de execução. O processo é concretizado através de métodos de diagramas hierárquicos ou árvores de funções [fig. A17].



[fig. A17] Árvore da função "tratamento da ordem de produção/ fornecimento"

As funções são representadas por rectângulos com vértices não rectos e podem ser particionados em vários níveis hierárquicos. As funções contidas no nível mais baixo da hierarquia, são denominadas de funções elementares. Para particionar uma função em sub-funções alguns critérios podem ser utilizados:

(i) aproximação temporal,

(ii) processam os mesmos objectos de informação, e
(iii) utilizam as mesmas operações.

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000042453